EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	34201	quer$6 near2 (database or (data adj base))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:26
L2	0	1 and depedency adj5 (code or graph)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:11
L3	222	1 and dependency adj5 (code or graph)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:26
L4	96	3 and recurs$6	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:11
L5	6	3 and recurs$6 near4 query$6	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:26
L6	408	1 and recurs$6 near4 (procedure or query$6 or path)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:16
L7	19	6 and dependency adj5 (code or graph)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:18
L8	32	6 and debug$6	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:17
L9	2	8 and dependency adj5 (code or graph)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:18

EAST Search History

L10	6	8 and dependency same (code or graph)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:21
L11	8	8 and dependency same (code or graph or debug$6)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:26
L12	2	"20010049682".pn.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:26
L13	1	12 and recurs$6 and query$6	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:26
L14	1	13 and (database or (data adj base))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:27
L16	1	12 and dependency and code and graph and procedure and debug$6 and path	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:27

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	494	quer$6 and procedure and recursive and path and graph and dependency	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:50
L2	609	quer$6 and procedure and recursive$2 and path and graph and dependency	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:33
L3	229	2 and debug$6	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:47
L4	106	3 and (database or (data adj base)) near2 quer$6	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:45
L5	5	4 and graph near dependency	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:34
L6	18	3 and graph near dependency	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:48
L7	24	4 and select! near9 (clause or statement or SQL)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:41
L8	11	4 and select! near9 (clause or statement or SQL) same quer$6	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:46
L9	11	8 and quer$6 adj4 (database or (data adj base))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:46

EAST Search History

L10	79	4 and quer$6 adj4 (database or (data adj base))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:47
L11	11	10 and select! near9 (clause or statement or SQL) same quer$6	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:47
L12	247	2 and quer$6 adj4 (database or (data adj base))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:52
L13	37	12 and select! near9 (clause or statement or SQL) same quer$6	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:47
L14	32	12 and debug$6 same code	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:50
L15	1	14 and dependency adj3 graph	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:48
L16	10832	debug$6 same code	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:50
L17	5	16 and (quer$6 same (procedure and recursive$2)) and graph and dependency	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:51
L18	52	16 and (quer$6 same recursive$2) and graph and dependency	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:51
L19	1	16 and (quer$6 same recursive$2) and (graph near3 dependency)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:51

EAST Search History

L20	1	16 and (quer$6 same recursive$2) and (graph near dependency)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:52
L21	117	16 and (quer$6 same recursive$2)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:52
L22	54	16 and (quer$6 with recursive$2)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:52
L23	27	22 and quer$6 adj4 (database or (data adj base))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:53

EAST Search History

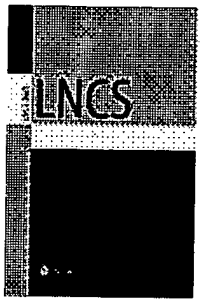
Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	494	quer$6 and procedure and recursive and path and graph and dependency	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:50
L2	609	quer$6 and procedure and recursive$2 and path and graph and dependency	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:33
L3	229	2 and debug$6	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:47
L4	106	3 and (database or (data adj base)) near2 quer$6	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:45
L5	5	4 and graph near dependency	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:34
L6	18	3 and graph near dependency	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:48
L7	24	4 and select! near9 (clause or statement or SQL)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:41
L8	11	4 and select! near9 (clause or statement or SQL) same quer$6	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:46
L9	11	8 and quer$6 adj4 (database or (data adj base))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:46

EAST Search History

L10	79	4 and quer$6 adj4 (database or (data adj base))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:47
L11	11	10 and select! near9 (clause or statement or SQL) same quer$6	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:47
L12	247	2 and quer$6 adj4 (database or (data adj base))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:52
L13	37	12 and select! near9 (clause or statement or SQL) same quer$6	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:47
L14	32	12 and debug$6 same code	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:50
L15	1	14 and dependency adj3 graph	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:54
L16	10832	debug$6 same code	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:50
L17	5	16 and (quer$6 same (procedure and recursive$2)) and graph and dependency	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:51
L18	52	16 and (quer$6 same recursive$2) and graph and dependency	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:51
L19	1	16 and (quer$6 same recursive$2) and (graph near3 dependency)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:51

EAST Search History

L20	1	16 and (quer$6 same recursive$2) and (graph near dependency)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:52
L21	117	16 and (quer$6 same recursive$2)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:52
L22	54	16 and (quer$6 with recursive$2)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:52
L23	27	22 and quer$6 adj4 (database or (data adj base))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:53
L25	458	16 and quer$6 adj4 (database or (data adj base))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:55
L26	10	25 and dependency same graph	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:55
L27	34130	quer$6 adj4 (database or (data adj base))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:55
L28	68	27 and (dependency or debug$6) same (graph and code)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:57
L29	27	28 and procedure same object	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:58
L30	27	28 and procedure same (graph or depedenc$3)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/15 08:59



Implication of Functional Dependencies for Recursive Queries

Book Series	Lecture Notes in Computer Science
Publisher	Springer Berlin / Heidelberg
ISSN	0302-9743 (Print) 1611-3349 (Online
Volume	Volume 2890/2003
Book	Perspectives of System Informatics
DOI	10.1007/b94823
Copyright	2003
ISBN	978-3-540-20813-6
Category	Databases
Pages	509-519
Subject Collection	Computer Science
SpringerLink Date	Thursday, February 19, 2004

José R. Paramá[1], Nieves R. Brisaboa[1], Miguel R. Penabad[1] and Ángeles S. Places[1]

(1) Database Lab. Computer Science Dept. Univ. of A Coruña. Campus de Elviña s/n, 15071 A Coruña. Spain. Tf. +34981-167000. Fax. +34981-167160,

After two decades of research in Deductive Databases, SQL99 [12] brings them again to the foreground given that SQL99 includes queries with linear recursion. Therefore some of the problems solved for the relational model demand our attention again.

In this paper, we tackle the *implication of functional dependencies* (also known as the *FD-FD implication problem*) in the deductive model framework. The problem is as follows. *Given P, F, and f, where P is a Datalog program, F is a set of functional dependencies defined on the predicates of P, and f is a fd defined over the predicates of P, is it true that for all databases d defined exclusively on the extensional predicates of P, d satisfies F implies that P (d) –the output database– satisfies f.* Unlike the implication problem of functional dependencies in the relational data model, this problem is undecidable for general Datalog programs.

In this paper, we provide two methods to check if a given set of fds will be satisfied by the output database (without computing

such database) for a class of Datalog programs.

Web Search Results for "recursive query dependency procedur..."

Search Filter: Moderate

1 - 20 of 50 from All Search Engines (About Results)

1. On the decidability and finite controllability of query processing ...
   The kinds of ICs that we consider are functional dependencies (in particular key ... Recursive query plans for data integration. J. of Logic Programming, ...
   portal.acm.org/citation.cfm?id=1142351.1142404
   [Found on Google]

2. SYSTEM AND METHOD FOR RECURSIVE PATH ANALYSIS OF DBMS PROCEDURES - Patent 20010049...
   ... and then for each dependency found, doing the query recursively until all basic ... for the procedure that is being debugged and all its dependent procedures. ...
   www.freepatentsonline.com/20010049682.html
   [Found on Yahoo! Search, Ask.com]

3. tcpdump - Linux Command - Unix Command
   Linux / Unix Command Library: tcpdump. Learn about its synopsis, description, options, and examples.
   linux.about.com/library/cmd/blcmdl8_tcpdump.htm
   [Found on About]

4. SYSTEM AND METHOD FOR RECURSIVE PATH ANALYSIS OF DBMS PROCEDURES ...
   Using the recursive procedure in Step 1 to determine the dependencies of implementations of object oriented code objects, based on a query against the ...
   www.freepatentsonline.com/EP1208459.html
   [Found on Google]

5. Oracle Dependency Management
   ... a query that references tables or other views; a procedure's body can ... procedure is invalidated if any timestamps of remotely referenced procedures do ...
   www.csee.umbc.edu/help/oracle8/server.815/a67781/c...
   [Found on Yahoo! Search]

6. Citations: A Generalization of the Differential Approach to ...
   Balbin, I. and Ramamohanarao, K. A Generalization of the Differential Approach to Recursive Query Relations. Journal of Logic Programming to ...
   citeseer.comp.nus.edu.sg/context/158889/0
   [Found on Ask.com]

7. 6 Dependencies Among Schema Objects
   ... time stamp checking dependency model, whenever a procedure is compiled or ... all other local procedures that depend on the remote procedure with the new time ...
   www.stanford.edu/dept/itss/docs/oracle/10g/server....
   [Found on Yahoo! Search]

Are you looking for?
No Suggestions Found.

8. perl58delta - Linux Command - Unix Command
   Linux / Unix Command Library: perl58delta. Learn about its synopsis, description, options, and examples.
   linux.about.com/library/cmd/blcmdl1_perl58delta.ht...
   [Found on About]

9. Query Reformulation in the Presence of Functional Dependencies ...
   We first analyse the unification **procedure** of the MiniCon algorithm that plays ..... DUSCHKA, O.M., GENESERETH, M.R., LEVY, A.Y., **Recursive Query** Plans for ...
   www.cs.qub.ac.uk/~J.Hong/iiwas03.doc [Found on Google]

10. Citations: An amateur's introduction to recursive query-processing ...
    F. Bancilhon, and R. Ramakrishnan. An amateur's introduction to **recursive** query-processing strategies. In ACM SIGMOD, pages 16-52, 1986.
    citeseer.ist.psu.edu/context/37532/0 [Found on Ask.com]

11. LNCS 2890 - Implication of Functional Dependencies for Recursive ...
    method than computing the **recursive query** and after that checking whether the ... A proof **procedure** for data dependencies. J. ACM,. 31:718□741, 1984. ...
    www.springerlink.com/index/0duykgprd9jx1v0n.pdf
    [Found on Google]

12. Dependencies Among Schema Objects
    ... time stamp checking **dependency** model, whenever a **procedure** is compiled or ... all other local procedures that depend on the remote **procedure** with the new time ...
    download-west.oracle.com/docs/cd/B10501_01/server....
    [Found on Yahoo! Search]

13. Citations: Evaluation of Recursive Queries Using Join Indices - ...
    Optimization Strategies for Parallel Linear **Recursive** Query.. ... techniques and operators for **recursive query** processing, e.g. in [Agr87] ...
    citeseer.ist.psu.edu/context/74338/0 [Found on Ask.com]

14. Resolving circular dependency between view and procedure
    Resolving circular **dependency** between view and **procedure** ... **Query** on dblink returning ORA-12545 error ...
    whatis.techtarget.com/expert/KnowledgebaseAnswer/0...
    [Found on Ask.com]

15. Chaotic fixpoint iteration guided by dynamic dependency
    a side-effect modifies the dynamic **dependency** information. **Procedure** .... **recursive query** evaluation. J. Logic Programming 1987, 4, pp. 259-262. ...
    www.springerlink.com/index/7k7u8436258135t2.pdf
    [Found on Google]

16. Query Rewriting Using Views in the Presence of Inclusion Dependencies
    dependency holds: procedure(procedure_name, patient_name) event(event_name, patient_name) ... A. Y. Levy, Recursive. **Query** Plans for Data Integration. ...
    delivery.acm.org/10.1145/960000/956729/p134-bai.pd...
    [Found on Yahoo! Search]

17. **Resolving circular dependency between view and procedure**
Resolving circular **dependency** between view and **procedure** ... 00604: error occurred at **recursive** level 1" and "ORA-01000: maximum open cursors ... searchoracle.techtarget.com/expert/KnowledgebaseAn... [Found on Ask.com]

18. **Capturing both types and integrity constraints in data integration**
DTD-conformance: **recursive**, nondeterministic; integrity constraints: validation .... scheduling: given a fixed **query dependency** graph, a heuristic based on ... homepages.inf.ed.ac.uk/wenfei/talk/sigmod03.ppt [Found on Google]

19. **dbTalk Databases Forums - View Single Post - Using variables in ...**
On Fri, 22 Jul 2005 01:06:01 -0700, Dunner wrote: Hi Hugo, What I am trying to do is write a **recursive query** to find the dependencies of a list of... www.dbtalk.net/578032-post4.html [Found on Ask.com]

20. **Divide-and-query and Subterm Dependency Tracking in the Mercury Declarative Debugg...**
procedures, even if they represent di erent modes of the ... uni cation) in that **procedure** makes that variable ground. ... one of the two **recursive** calls, ... www.cs.mu.oz.au/research/mercury/information/paper... [Found on Yahoo! Search]

1 | 2 | 3 Next >

Web Search Results for "recursive query dependency procedur..."

Search Filter: Moderate

21. Microsoft Word - IJCAI-workshop-03.doc
... reformu-late the user query into recursive queries on the sources to satisfy binding patterns. ... way to write a recursive query plan ...
www.isi.edu/~thakkar/IJCAI-03-ws.pdf [Found on Ask.com]

22. Efficient Execution of Recursive Integration Plans
late the user query into recursive queries on the sources. to satisfy binding patterns. ... view definition, we can use a flattening procedure described in ...
www.isi.edu/integration/papers/thakkar03-iiweb.pdf
[Found on Yahoo! Search]

23. PODS 1987: 349-362
A procedure for testing uniform equivalence is also developed for the case in which ... An Amateur's Introduction to Recursive Query Processing Strategies. ...
www.informatik.uni-trier.de/~ley/db/conf/pods/Sagi...
[Found on Google]

24. The Kernel Recursive Least Squares Algorithm
This sparsi cation procedure is crucial to the operation of KRLS, as it ... (iii) their prediction (query) time often scales linearly with the training set size. ...
www.ee.technion.ac.il/~rmeir/Publications/KrlsRepo...
[Found on Yahoo! Search]

25. An Algorithm for Query Folding with Functional Dependencies
Recursive plans for information gathering. In. Proc. of 15th IJCAI. ,. Nagoya, Japan, August 1997. 5. J. Gryz. Query folding with inclusion dependencies. In ...
www.cse.yorku.ca/~jarek/papers/iis98/paper.ps
[Found on Google]

26. CISC 879 : Term Paper Query Languages
... lack of support for internationalization/localization, and platform-dependency. ... database and your own programs/procedures that manipulate it, but there ...
www.cis.udel.edu/~wchen/courses/cisc879/
[Found on Yahoo! Search]

27. Amazon.com: "procedure permute": Key Phrase page
Key Phrase page for procedure permute: Books containing the phrase procedure permute...
amazon.com/phrase/procedure-permute
[Found on Ask.com]

28. Domain name system - Wikipedia, the free

Are you looking for?

No Suggestions Found.

Recent Searches

recursive query depend...

Hide Clear All

encyclopedia
... resolver) negotiate use of recursive service using bits in the query headers. ... by segment, from right to left, using an iterative search procedure. ...
en.wikipedia.org/wiki/Domain_name_system
[Found on Yahoo! Search]

29. Bookpool: An Introduction to Database Systems
Join dependencies and fifth normal form. The normalization procedure summarized. ... Recursive query processing. VI. OBJECTS, RELATIONS, AND XML. ...
www.bookpool.com/sm/0321197844 [Found on Google]

30. Amazon.com: "chain query": Key Phrase page
Key Phrase page for chain query: Books containing the phrase chain query ... ... approximate dependency inference, incremental query ...
amazon.com/phrase/chain-query [Found on Ask.com]

31. Principles of Parallel Algorithm Design
Consider the task dependency graphs of the two database. query decompositions: 10. 10 ... procedure RECURSIVE MIN (A, n) 2. begin. 3. if (n = 1) then. 4. ...
www-users.cs.umn.edu/~karypis/parbook/Lectures/AG/...
[Found on Yahoo! Search]

32. ppt
Consider the task dependency graphs of the two database query. decompositions: ... Recursive Decomposition ... 1. procedure SERIAL_MIN (A, n) 2. begin. 3. ...
www-users.cs.umn.edu/~karypis/parbook/Lectures/AG/...
[Found on Yahoo! Search]

33. The Kernel Recursive Least-Squares Algorithm
sification procedure as an approximate online form of principal ... memory are consumed in the operational (query) stage of the ...
www.cs.ualberta.ca/~yaki/papers/ieee_krls.pdf
[Found on Yahoo! Search]

34. Propagator: A Family of Patterns
... dependency network ... the same procedure recursively to. determine whether they are out of date. After an object finds it is out of date, it may query ...
www.sei.cmu.edu/pub/documents/articles/pdf/propaga...
[Found on Yahoo! Search]

35. Question Answering for Dutch using Dependency Relations
recursive base constituents), named entity recognition, and syntactic or ... the effectivity of question classi cation and query expansion based on ontologi ...
odur.let.rug.nl/~gosse/Imix/project_description.pd...
[Found on Yahoo! Search]

36. Capturing both Types and Constraints in Data Integration
scribing treatment procedures a treatment may require a ... can handle recursive DTDs and ... xed during optimization, the query dependency graph G is ...
www.cs.utah.edu/~juliana/pub/sigmod2003.pdf
[Found on Yahoo! Search]

37. An Active Conceptual Model for Fixed Income Securities Analysis for Multiple Finan...
... from the system design stage to providing query access to ... is a recursive procedure that takes a

reference object and a path and follows a depth ...
web.mit.edu/amoulton/www/er98.pdf
[Found on Yahoo! Search]

38. **Determining Service Dependencies in Distributed Systems**
form interface to query service and dependency information ... shown that queries for (recursive) drill up or drill down opera ...
www.research.ibm.com/people/a/akeller/Data/icc2001...
[Found on Yahoo! Search]

39. **Estimating the Computational Cost of Logic Programs**
for recursive procedures are obtained in the form of di erence equations To get ... guments of the procedure For recursive clauses this yields a di erence ...
www.cs.arizona.edu/~debray/Publications/sas94.pdf
[Found on Yahoo! Search]

40. **Transparent access to multiple bioinformatics information sources**
... is used both to drive a visual query interface and as a global schema against ... strictly ordered chains of retrieval, filtering, and processing procedures. ...
www.research.ibm.com/journal/sj/402/goble.html
[Found on Yahoo! Search]

< Prev 1 | 2 | 3 Next >

Web Search Results for "recursive query dependency procedur..."

Search Filter: Moderate

41 - 50 of 50 from All Search Engines (About Results)

41. Generic Abstract Interpretation Algorithms for Prolog: Two Optimization Techniques...
... to the original algorithm: **dependency** on clause pre xes and ... It is mainly a deterministic program, with mutually **recursive** procedures. The program ...
www.cs.ubc.ca/local/reading/proceedings/spe91-95/s...
[Found on Yahoo! Search]

42. SQL Manager | SQL Artikel | Alle SQL Artikel
... than the extended stored procedures available in earlier versions of ... include error handling, new **recursive query** capabilities, and support for new ...
www.sqlmanager.net/de/articles/655
[Found on Yahoo! Search]

43. Compile-Time Derivation of Variable Dependency Using Abstract Interpretation
... points in the execution of the program, rather than just at a **procedure** level. ... (least burden on the programmer) such **query** forms can be simply the names ...
clip.dia.fi.upm.es/papers/ai-jlp.pdf [Found on Yahoo! Search]

44. Using Partial Evaluation in Distributed Query Evaluation
There is a **dependency** relation be- tween partial evaluation processes for the **query** q on dif ... is essentially the QList(q) of the initial **query** q. **Recursive** ...
homepages.inf.ed.ac.uk/wenfei/papers/vldb06-p2p.pd...
[Found on Yahoo! Search]

45. SQL, PL/SQL, and Java
... to execute the **procedure** (but not the privileges to query, update, or delete ... "**Dependency** Tracking for Stored Procedures" ...
www.cs.uvm.edu/oracle9doc/server.901/a88856/c16sql...
[Found on Yahoo! Search]

46. Query Processing for Advanced Database Systems - Elsevier
... **Dependency** Graph 7.4 Type Checking 7.5 Conclusion 8 Tagging as ... about conditions of sale & ordering procedures, and links to our regional sales offices. ...
www.elsevier.com/wps/find/bookdescription.cws_home...
[Found on Yahoo! Search]

47. Reference Constraint Graph for Oracle, Informix
echo exit 1 } # **Query** for finding referential **dependency** pairs. ... **recursive** function tree_parent, will explode # harmlessly with circular graph. ...
www.tc.umn.edu/~hause011/code/ref_load_ord.ksh

Are you looking for?

No Suggestions Found.

Recent Searches

recursive query depend...

Hide Clear All

[Found on Yahoo! Search]

48. Data Exchange: Semantics and Query Answering
(the formalism used to express a target dependency may be different from those used ... of weakly recursive ILOG [12], even though ... the chase procedure. ...
www.cs.toronto.edu/db/dataexchange/icdt03.pdf
[Found on Yahoo! Search]

49. Improving Availability with Recursive Microreboots: A Soft-State System Case Study
forming an SQL query on a database and verifying the result provides reasonable ... which case the pre-recovery and post-recovery procedures are empty, checkpoint ...
swig.stanford.edu/~candea/papers/perfeval/perfeval...
[Found on Yahoo! Search]

50. Recursive Random Fields
procedure continues until no single-variable change will fur- ther improve the probability. ... We can compute this by iterating over all query predicates, ...
www.ijcai.org/papers07/Papers/IJCAI07-153.pdf
[Found on Yahoo! Search]

Terms used: **recursively query database** or **data base dependency graph code object procedure**

Found **33** of **214,158**

Sort results by relevance

Display results expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 33 Result page: **1** 2 next

Relevance scale

1 Cactis: a self-adaptive, concurrent implementation of an object-oriented database management system

Scott E. Hudson, Roger King

September 1989 **ACM Transactions on Database Systems (TODS)**, Volume 14 Issue 3

Publisher: ACM Press

Full text available: pdf(2.65 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Cactis is an object-oriented, multiuser DBMS developed at the University of Colorado. The system supports functionally-defined data and uses techniques based on attributed graphs to optimize the maintenance of functionally-defined data. The implementation is self-adaptive in that the physical organization and the update algorithms dynamically change in order to reduce disk access. The system is also concurrent. At any given time there are some number of computations that must be performed t ...

2 Cost-driven vertical class partitioning for methods in object oriented databases

Chi-Wai Fung, Kamalakar Karlapalem, Qing Li

October 2003 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 12 Issue 3

Publisher: Springer-Verlag New York, Inc.

Full text available: pdf(334.54 KB) Additional Information: full citation, abstract, citings, index terms

Abstract.In object-oriented databases (OODBs), a method encapsulated in a class typically accesses a few, but not all the instance variables defined in the class. It may thus be preferable to vertically partition the class for reducing irrelevant data (instance variables) accessed by the methods. Our prior work has shown that vertical class partitioning can result in a substantial decrease in the total number of disk accesses incurred for executing a set of applications, but coming up with an op ...

Keywords: Affinity-based, Analytical cost model, Cost-driven, Hill-climbing heuristic algorithm, Method-induced, Object-oriented databases, Vertical class partitioning

3 Queries and query processing in object-oriented database systems

David D. Straube, M. Tamer Özsu

October 1990 **ACM Transactions on Information Systems (TOIS)**, Volume 8 Issue 4

Publisher: ACM Press

Full text available: pdf(3.16 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Object-oriented database mangement systems (OODBMS) combine the data abstraction and computational models of object-oriented programming languages with the query and performance capabilities of database management systems. A concise, formal data model for OODBMS has not been universally accepted, preventing detailed investigation of various system issues such as query processing. We define a data model that captures the essence of classification-based object-oriented systems and formalize c ...

Keywords: object algebra, object calculus, object-oriented databases, query transformation rules

4 Logical foundations of object-oriented and frame-based languages

Michael Kifer, Georg Lausen, James Wu
July 1995 **Journal of the ACM (JACM)**, Volume 42 Issue 4
Publisher: ACM Press

Full text available: pdf(7.52 MB) Additional Information: full citation, abstract, references, citings, index terms, review

We propose a novel formalism, called Frame Logic (abbr., F-logic), that accounts in a clean and declarative fashion for most of the structural aspects of object-oriented and frame-based languages. These features include object identity, complex objects, inheritance, polymorphic types, query methods, encapsulation, and others. In a sense, F-logic stands in the same relationship to the object-oriented paradigm as classical predicate calculus stands to relational programming. ...

Keywords: deductive databases, frame-based languages, logic programming, nonmonotonic inheritance, object-oriented programming, proof theory, semantics, typing

5 Programming constructs for database system implementation in EXODUS

Joel E. Richardson, Michael J. Carey
December 1987 **ACM SIGMOD Record , Proceedings of the 1987 ACM SIGMOD international conference on Management of data SIGMOD '87**, Volume 16 Issue 3
Publisher: ACM Press

Full text available: pdf(1.60 MB) Additional Information: full citation, abstract, references, citings, index terms

The goal of the EXODUS extensible DBMS project is to enable the rapid development of a wide spectrum of high-performance, application-specific database systems EXODUS provides certain kernel facilities for use by all applications and a set of tools to aid the database implementor (DBI) in generating new database system software. Some of the DBI's work is supported by EXODUS tools which generate database components from a specification. However, components such as new abstract data types, ac ...

6 Special issue on prototypes of deductive database systems: The CORAL deductive system

Raghu Ramakrishnan, Divesh Srivastava, S. Sudarshan, Praveen Seshadri
April 1994 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 3 Issue 2
Publisher: Springer-Verlag New York, Inc.

Full text available: pdf(3.03 MB) Additional Information: full citation, abstract, references, citings

CORAL is a deductive system that supports a rich declarative language, and an interface to C++, which allows for a combination of declarative and imperative programming. A CORAL

declarative program can be organized as a collection of interacting modules. CORAL supports a wide range of evaluation strategies, and automatically chooses an efficient strategy for each module in the program. Users can guide query optimization by selecting from a wide range of control choices. The CORAL system provides ...

Keywords: deductive database, logic programming system, query language

7 The FINITE STRING newsletter: Abstracts of current literature

Computational Linguistics Staff
July 1984 **Computational Linguistics**, Volume 10 Issue 3-4
Publisher: MIT Press
Full text available: pdf(2.30 MB) Publisher Site Additional Information: full citation

8 Shape-based retrieval and analysis of 3D models

Thomas Funkhouser, Michael Kazhdan
August 2004 **ACM SIGGRAPH 2004 Course Notes SIGGRAPH '04**
Publisher: ACM Press
Full text available: pdf(12.56 MB) Additional Information: full citation, abstract

Large repositories of 3D data are rapidly becoming available in several fields, including mechanical CAD, molecular biology, and computer graphics. As the number of 3D models grows, there is an increasing need for computer algorithms to help people find the interesting ones and discover relationships between them. Unfortunately, traditional text-based search techniques are not always effective for 3D models, especially when queries are geometric in nature (e.g., find me objects that fit into thi ...

9 An axiomatic model of dynamic schema evolution in objectbase systems

Randel J. Peters, M. Tamer Özsu
March 1997 **ACM Transactions on Database Systems (TODS)**, Volume 22 Issue 1
Publisher: ACM Press
Full text available: pdf(647.83 KB) Additional Information: full citation, references, citings, index terms

Keywords: dynamic schema evolution, object database management systems

10 Exchanging intensional XML data

Tova Milo, Serge Abiteboul, Bernd Amann, Omar Benjelloun, Fred Dang Ngoc
March 2005 **ACM Transactions on Database Systems (TODS)**, Volume 30 Issue 1
Publisher: ACM Press
Full text available: pdf(1.07 MB) Additional Information: full citation, abstract, references, index terms

XML is becoming the universal format for data exchange between applications. Recently, the emergence of Web services as standard means of publishing and accessing data on the Web introduced a new class of XML documents, which we call *intensional* documents. These are XML documents where some of the data is given explicitly while other parts are defined only intensionally by means of embedded calls to Web services.When such documents are exchanged between applications, one has the choice of ...

Keywords: Data exchange, Web services, XML, intensional information, typing

11 GENOA—a customizable, front-end-retargetable source code analysis framework

Premkumar T. Devanbu

April 1999 **ACM Transactions on Software Engineering and Methodology (TOSEM)**, Volume 8 Issue 2

Publisher: ACM Press

Full text available: pdf(241.27 KB) Additional Information: full citation, abstract, references, citings, index terms, review

Code analysis tools provide support for such software engineering tasks as program understanding, software metrics, testing, and reengineering. In this article we describe GENOA, the framework underlying application generators such as Aria and GEN++ which have been used to generate a wide range of practical code analysis tools. This experience illustrates front-end retargetability of GENOA; we describe the features of the GENOA framework that allow it to be ...

Keywords: code inspection, metrics, reverse engineering, source analysis

12 Parallel execution of prolog programs: a survey

Gopal Gupta, Enrico Pontelli, Khayri A.M. Ali, Mats Carlsson, Manuel V. Hermenegildo

July 2001 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 23 Issue 4

Publisher: ACM Press

Full text available: pdf(1.95 MB) Additional Information: full citation, abstract, references, citings, index terms

Since the early days of logic programming, researchers in the field realized the potential for exploitation of parallelism present in the execution of logic programs. Their high-level nature, the presence of nondeterminism, and their referential transparency, among other characteristics, make logic programs interesting candidates for obtaining speedups through parallel execution. At the same time, the fact that the typical applications of logic programming frequently involve irregular computatio ...

Keywords: Automatic parallelization, constraint programming, logic programming, parallelism, prolog

13 Attribute grammar paradigms—a high-level methodology in language implementation

Jukka Paakki

June 1995 **ACM Computing Surveys (CSUR)**, Volume 27 Issue 2

Publisher: ACM Press

Full text available: pdf(5.15 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Attribute grammars are a formalism for specifying programming languages. They have been applied to a great number of systems automatically producing language implementations from their specifications. The systems and their specification languages can be evaluated and classified according to their level of application support, linguistic characteristics, and degree of automation.A survey of attribute grammar-based specification languages is given. The modern advanced specification ...

Keywords: attribute grammars, blocks, classes, compiler writing systems, functional dependencies, incomplete data, incrementality, inheritance, language processing, language processor generators, lazy evaluation, logical variables, objects, parallelism, processes, programming paradigms, semantic functions, symbol tables, unification

14 Data integration and sharing I: Capturing both types and constraints in data integration

Michael Benedikt, Chee-Yong Chan, Wenfei Fan, Juliana Freire, Rajeev Rastogi

June 2003 **Proceedings of the 2003 ACM SIGMOD international conference on Management of data SIGMOD '03**

Publisher: ACM Press

Full text available: pdf(690.62 KB) Additional Information: full citation, abstract, references, citings, index terms

We propose a framework for integrating data from multiple relational sources into an XML document that both conforms to a given DTD and satisfies predefined XML constraints. The framework is based on a specification language, AIG, that extends a DTD by (1) associating element types with semantic attributes (inherited and synthesized, inspired by the corresponding notions from Attribute Grammars), (2) computing these attributes via parameterized SQL queries over multiple data sources, and (3) inc ...

15 Incremental computation of complex object queries

Hiroaki Nakamura

October 2001 **ACM SIGPLAN Notices , Proceedings of the 16th ACM SIGPLAN conference on Object oriented programming, systems, languages, and applications OOPSLA '01**, Volume 36 Issue 11

Publisher: ACM Press

Full text available: pdf(226.82 KB) Additional Information: full citation, abstract, references, citings, index terms

The need for incremental algorithms for evaluating database queries is well known, but constructing algorithms that work on object-oriented databases (OODBs) has been difficult. The reason is that OODB query languages involve complex data types including composite objects and nested collections. As a result, existing algorithms have limitations in that the kinds of database updates are restricted, the operations found in many query languages are not supported, or the algorithms are too complex t ...

16 Graph rewrite systems for program optimization

Uwe Assmann

July 2000 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 22 Issue 4

Publisher: ACM Press

Full text available: pdf(571.22 KB) Additional Information: full citation, abstract, references, citings, index terms, review

Graph rewrite systems can be used to specify and generate program optimizations. For termination of the systems several rule-based criteria are developed, defining exhaustive graph rewrite systems. For nondeterministic systems stratification is introduced which automatically selects single normal forms. To illustrate how far the methodology reaches, parts of the lazy code motion optimization are specified. The resulting graph rewrite system classes can be e ...

Keywords: compiler generators, graph rewrite systems, program analysis, program optimization, program transformation, specification, stratification, very high-level languages, visual programming

17 Software evolution: Generating programming language-based pattern matchers

Santanu Paul, Atul Prakash

October 1993 **Proceedings of the 1993 conference of the Centre for Advanced Studies on Collaborative research: software engineering - Volume 1 CASCON '93**

Publisher: IBM Press

Full text available: pdf(1.55 MB) Additional Information: full citation, abstract, references, citings

This paper is based on a logical extension of our past work in pattern matching tools [22, 24, 25] for reverse engineering. We explore two new directions: first, we investigate the need for new and more powerful source code and pattern representations to support a richer set of queries; and second, we develop the concept of automatic generation of pattern matchers for different programming languages starting from a high-level specification of the programming language. A generator will eliminate ...

18 DB-1 (databases): data integration: Composable XML integration grammars

Wenfei Fan, Minos Garofalakis, Ming Xiong, Xibei Jia

November 2004 **Proceedings of the thirteenth ACM international conference on Information and knowledge management CIKM '04**

Publisher: ACM Press

Full text available: pdf(257.14 KB) Additional Information: full citation, abstract, references, index terms

The proliferation of XML as a standard for data representation and exchange in diverse, next-generation Web applications has created an emphatic need for effective XML data-integration tools. For several real-life scenarios, such XML data integration needs to be DTD-directed -- in other words, the target, integrated XML database must conform to a prespecified, user- or application-defined DTD. In this paper, we propose a novel formalism, XML Integration Grammars (XIGs)&# ...

Keywords: XML, data integration, grammar

19 The elements of nature: interactive and realistic techniques

Oliver Deusen, David S. Ebert, Ron Fedkiw, F. Kenton Musgrave, Przemyslaw Prusinkiewicz, Doug Roble, Jos Stam, Jerry Tessendorf

August 2004 **ACM SIGGRAPH 2004 Course Notes SIGGRAPH '04**

Publisher: ACM Press

Full text available: pdf(17.65 MB) Additional Information: full citation, abstract

This updated course on simulating natural phenomena will cover the latest research and production techniques for simulating most of the elements of nature. The presenters will provide movie production, interactive simulation, and research perspectives on the difficult task of photorealistic modeling, rendering, and animation of natural phenomena. The course offers a nice balance of the latest interactive graphics hardware-based simulation techniques and the latest physics-based simulation techni ...

20 Design technologies: Theories and techniques of program understanding

Santanu Paul, Atul Prakash, Erich Buss, John Henshaw

October 1991 **Proceedings of the 1991 conference of the Centre for Advanced Studies on Collaborative research CASCON '91**

Publisher: IBM Press

Full text available: pdf(1.28 MB) Additional Information: full citation, abstract, references, citings

Understanding programs written by others is a difficult task. Most maintenance tasks in software require an understanding of the source code as a first step. Unfortunately, in most cases, the source code of a program is the only reliable documentation of its behavior. As a result, maintainers have to rely heavily on code browsing to acquire the necessary information. This paper surveys the various theories that have been proposed to explain the process of understanding, considers the different t ...

Results 21 - 33 of 33 Result page: previous 1 2

Relevance scale

21 Effectivness of abstract interpretation in automatic parallelization: a case study in logic programming

Francisco Bueno, María García de la Banda, Manuel Hermenegildo

March 1999 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 21 Issue 2

Publisher: ACM Press

Full text available: pdf(533.48 KB) Additional Information: full citation, abstract, references, citings, index terms

We report on a detailed study of the application and effectiveness of program analysis based on abstract interpretation of automatic program parallelization. We study the case of parallelizing logic programs using the notion of strict independence. We first propose and prove correct a methodology for the application in the parallelization task of the information inferred by abstract interpretation, using a parametric domain. The methodology is generic in the sense of allowing the use of dif ...

Keywords: abstract interpretation, automatic parallelization, data flow analysis, logic programming, parallelism

22 Supporting multiple view maintenance policies

Latha S. Colby, Akira Kawaguchi, Daniel F. Lieuwen, Inderpal Singh Mumick, Kenneth A. Ross

June 1997 **ACM SIGMOD Record , Proceedings of the 1997 ACM SIGMOD international conference on Management of data SIGMOD '97**, Volume 26 Issue 2

Publisher: ACM Press

Full text available: pdf(1.55 MB) Additional Information: full citation, abstract, references, citings, index terms

Materialized views and view maintenance are becoming increasingly important in practice. In order to satisfy different data currency and performance requirements, a number of view maintenance policies have been proposed. Immediate maintenance involves a potential refresh of the view after every update to the deriving tables. When staleness of views can be tolerated, a view may be refreshed periodically or (on-demand) when it is queried. The maintenance policies that are chosen for views hav ...

23 Resolving non-uniqueness in design feature histories

Vincent A. Cicirello, William C. Regli

June 1999 **Proceedings of the fifth ACM symposium on Solid modeling and**

applications SMA '99
Publisher: ACM Press
Full text available: pdf(1.22 MB) Additional Information: full citation, references, citings, index terms

Keywords: computer-aided design, constraint-based and parametric modeling, engineering knowledge-bases, feature-based modeling, modeling families of geometric objects, product data management

24 Flexible access control policy specification with constraint logic programming
Steve Barker, Peter J. Stuckey
November 2003 **ACM Transactions on Information and System Security (TISSEC)**, Volume 6 Issue 4
Publisher: ACM Press
Full text available: pdf(421.66 KB) Additional Information: full citation, abstract, references, citings, index terms

We show how a range of role-based access control (RBAC) models may be usefully represented as constraint logic programs, executable logical specifications. The RBAC models that we define extend the "standard" RBAC models that are described by Sandhu et al., and enable security administrators to define a range of access policies that may include features, like denials of access and temporal authorizations, that are often useful in practice, but which are not widely supported in existing access co ...

Keywords: Role-based access control, constraint logic programming

25 Controlling generalization and polyvariance in partial deduction of normal logic programs
Michael Leuschel, Bern Martens, Danny De Schreye
January 1998 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 20 Issue 1
Publisher: ACM Press
Full text available: pdf(911.56 KB) Additional Information: full citation, abstract, references, citings, index terms, review

Given a program and some input data, partial deduction computes a specialized program handling any remaining input more efficiently. However, controlling the process well is a rather difficult problem. In this article, we elaborate global control for partial deduction: for which atoms, among possibly infinitely many, should specialized relations be produced, meanwhile guaranteeing correctness as well as termination? Our work is based on two ingredients. First, we use the concept of a ch ...

Keywords: flow analysis, partial deduction, partial evaluation, program transformation, supercompilation

26 Reports: Research in database engineering at the University of Namur
Jean-Luc Hainaut
December 2003 **ACM SIGMOD Record**, Volume 32 Issue 4
Publisher: ACM Press
Full text available: pdf(151.57 KB) Additional Information: full citation, references

27

DTD inference for views of XML data
Yannis Papakonstantinou, Victor Vianu
May 2000 **Proceedings of the nineteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems PODS '00**
Publisher: ACM Press
Full text available: pdf(347.61 KB) Additional Information: full citation, abstract, references, citings, index terms

We study the inference of Data Type Definitions (DTDs) for views of XML data, using an abstraction that focuses on document content structure. The views are defined by a query language that produces a list of documents selected from one or more input sources. The selection conditions involve vertical and horizontal navigation, thus querying explicitly the order present in input documents. We point several strong limitations in the descriptive ability of current DTDs and the need for extendi ...

28 Unfolding partiality and disjunctions in stable model semantics
Tomi Janhunen, Ilkka Niemelä, Dietmar Seipel, Patrik Simons, Jia-Huai You
January 2006 **ACM Transactions on Computational Logic (TOCL)**, Volume 7 Issue 1
Publisher: ACM Press
Full text available: pdf(505.24 KB) Additional Information: full citation, abstract, references, index terms

This article studies an implementation methodology for partial and disjunctive stable models where partiality and disjunctions are unfolded from a logic program so that an implementation of stable models for normal (disjunction-free) programs can be used as the core inference engine. The unfolding is done in two separate steps. First, it is shown that partial stable models can be captured by total stable models using a simple linear and modular program transformation. Hence, reasoning tasks conc ...

Keywords: Answer set programming, disjunctive stable models, inference engine, minimal models, partial models, quantified Boolean formulas

29 The specification and enforcement of authorization constraints in workflow management systems
Elisa Bertino, Elena Ferrari, Vijay Atluri
February 1999 **ACM Transactions on Information and System Security (TISSEC)**, Volume 2 Issue 1
Publisher: ACM Press
Full text available: pdf(374.02 KB) Additional Information: full citation, abstract, references, citings, index terms, review

In recent years, workflow management systems (WFMSs) have gained popularity in both research and commercial sectors. WFMSs are used to coordinate and streamline business processes. Very large WFMSs are often used in organizations with users in the range of several thousands and process instances in the range of tens and thousands. To simplify the complexity of security administration, it is common practice in many businesses to allocate a role for each activity in the process and then assig ...

Keywords: access control, authorization constraints, role and user planning

30 abc: an extensible AspectJ compiler
Pavel Avgustinov, Aske Simon Christensen, Laurie Hendren, Sascha Kuzins, Jennifer Lhoták, Ondřej Lhoták, Oege de Moor, Damien Sereni, Ganesh Sittampalam, Julian Tibble
March 2005 **Proceedings of the 4th international conference on Aspect-oriented software development AOSD '05**
Publisher: ACM Press

Full text available: pdf(128.27 KB) Additional Information: full citation, abstract, references, citings

Research in the design of aspect-oriented programming languages requires a workbench that facilitates easy experimentation with new language features and implementation techniques. In particular, new features for AspectJ have been proposed that require extensions in many dimensions: syntax, type checking and code generation, as well as data flow and control flow analyses.The AspectBench Compiler *(abc)* is an implementation of such a workbench. The base version of *abc* implements the ...

31 Relational transducers for electronic commerce

Serge Abiteboul, Victor Vianu, Brad Fordham, Yelena Yesha

May 1998 **Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems PODS '98**

Publisher: ACM Press

Full text available: pdf(1.27 MB) Additional Information: full citation, references, citings, index terms

32 Data integration and sharing I: Exchanging intensional XML data

Tova Milo, Serge Abiteboul, Bernd Amann, Omar Benjelloun, Fred Dang Ngoc

June 2003 **Proceedings of the 2003 ACM SIGMOD international conference on Management of data SIGMOD '03**

Publisher: ACM Press

Full text available: pdf(237.21 KB) Additional Information: full citation, abstract, references, citings, index terms

XML is becoming the universal format for data exchange between applications. Recently, the emergence of Web services as standard means of publishing and accessing data on the Web introduced a new class of XML documents, which we call *intensional* documents. These are XML documents where some of the data is given explicitly while other parts are defined only intensionally by means of embedded calls to Web services.When such documents are exchanged between applications, one has the choice to ...

33 Flow equations as a generic programming tool for manipulation of attributed graphs

John Fiskio-Lasseter, Michal Young

November 2002 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 2002 ACM SIGPLAN-SIGSOFT workshop on Program analysis for software tools and engineering PASTE '02**, Volume 28 Issue 1

Publisher: ACM Press

Full text available: pdf(143.27 KB) Additional Information: full citation, abstract, references, citings, index terms

The past three decades have seen the creation of several tools that extract, visualize, and manipulate graph-structured representations of program information. To facilitate interconnection and exchange of information between these tools, and to support the prototyping and development of new tools, it is desirable to have some generic support for the specification of graph transformations and exchanges between them.GenSet is a generic programmable tool for transformation of gra ...

Results for "(((recusively)<in>metadata) <and> ((query)<in>metadata))<and> ((database..."
Your search matched **0** of **1685772** documents.
A maximum of **100** results are displayed, **25** to a page, sorted by **Relevance** in **Descending** order.

No results were found.

Please edit your search criteria and try again. Refer to the Help pages if you need assistance revising your search

Results for "(((querying)<in>metadata) <and> ((database)<in>metadata))<and> ((depend..."
Your search matched 92 of 1685772 documents.
A maximum of 100 results are displayed, 25 to a page, sorted by Relevance in Descending order.

1. **Mining approximate dependency to answer null queries on similarity-based fuzzy relationa**
Shyue-Liang Wang; Tzung-Pei Hong;
Fuzzy Systems, 2000. FUZZ IEEE 2000. The Ninth IEEE International Conference on
Volume 2, 7-10 May 2000 Page(s):615 - 620 vol.2
Digital Object Identifier 10.1109/FUZZY.2000.839063

2. **Update operation and partition dependency in distributed database systems**
Hao Chen;
Advanced Information Networking and Applications, 2003. AINA 2003. 17th International Confere
27-29 March 2003 Page(s):624 - 627
Digital Object Identifier 10.1109/AINA.2003.1192958

3. **Fuzzy functional dependency and its application to approximate data querying**
Intan, R.; Mukaidono, M.;
Database Engineering and Applications Symposium, 2000 International
18-20 Sept. 2000 Page(s):47 - 54
Digital Object Identifier 10.1109/IDEAS.2000.880561

4. **Towards the reverse engineering of renormalized relational databases**
Petit, J.-M.; Toumani, F.; Boulicaut, J.-F.; Kouloumdjian, J.;
Data Engineering, 1996. Proceedings of the Twelfth International Conference on
26 Feb.-1 March 1996 Page(s):218 - 227
Digital Object Identifier 10.1109/ICDE.1996.492110

5. **Design and evaluation of data allocation algorithms for distributed multimedia database sy**

Yu-Kwong Kwok; Karlapalem, K.; Ahmad, I.; Ng Moon Pun;
Selected Areas in Communications, IEEE Journal on
Volume 14, Issue 7, Sept. 1996 Page(s):1332 - 1348
Digital Object Identifier 10.1109/49.536483

AbstractPlus | References | Full Text: PDF(1652 KB) IEEE JNL
Rights and Permissions

6. **The study of rough relational database based on granular computing**
Qiusheng An; Yusheng Zhang; WenXiu Zhang;
Granular Computing, 2005 IEEE International Conference on
Volume 1, 25-27 July 2005 Page(s):108 - 111 Vol. 1
Digital Object Identifier 10.1109/GRC.2005.1547245

AbstractPlus | Full Text: PDF(968 KB) IEEE CNF
Rights and Permissions

7. **Automatic aggregation using explicit metadata**
Grumbach, S.; Tininini, L.;
Scientific and Statistical Database Management, 2000. Proceedings. 12th International Conferenc
26-28 July 2000 Page(s):85 - 94
Digital Object Identifier 10.1109/SSDM.2000.869780

AbstractPlus | Full Text: PDF(256 KB) IEEE CNF
Rights and Permissions

8. **Semantic query optimization using inter-relational functional dependencies**
Godin, R.; Missaoui, R.;
System Sciences, 1991. Proceedings of the Twenty-Fourth Annual Hawaii International Conferen
Volume iii, 8-11 Jan. 1991 Page(s):368 - 375 vol.3
Digital Object Identifier 10.1109/HICSS.1991.184165

AbstractPlus | Full Text: PDF(584 KB) IEEE CNF
Rights and Permissions

9. **Domain reduction dependencies: A new type of dependency for statistical database securi**
Hansen, S.C.; Unger, E.A.;
Computer Security Applications Conference, 1991. Proceedings., Seventh Annual
2-6 Dec. 1991 Page(s):178 - 186
Digital Object Identifier 10.1109/CSAC.1991.213007

AbstractPlus | Full Text: PDF(680 KB) IEEE CNF
Rights and Permissions

10. **A distributed query processing strategy using placement dependency**
Chengwen Liu; Hao Chen; Krueger, W.;
Data Engineering, 1996. Proceedings of the Twelfth International Conference on
26 Feb.-1 March 1996 Page(s):477 - 484
Digital Object Identifier 10.1109/ICDE.1996.492197

AbstractPlus | Full Text: PDF(776 KB) IEEE CNF
Rights and Permissions

11. **Efficiently Detecting Inclusion Dependencies**
Bauckmann, J.; Leser, U.; Naumann, F.; Tietz, V.;
Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on
15-20 April 2007 Page(s):1448 - 1450
Digital Object Identifier 10.1109/ICDE.2007.369032

AbstractPlus | Full Text: PDF(84 KB) IEEE CNF
Rights and Permissions

12. **Automatic relationship discovery in self-managing database systems**
Ilyas, I.; Markl, V.; Haas, P.; Brown, P.G.; Aboulnaga, A.;
Autonomic Computing, 2004. Proceedings. International Conference on
17-18 May 2004 Page(s):340 - 341

Digital Object Identifier 10.1109/ICAC.2004.1301405

AbstractPlus | Full Text: PDF(1341 KB) IEEE CNF
Rights and Permissions

13. **A framework for understanding existing databases**
Lopes, S.; Petit, J.-M.; Lakhal, L.;
Database Engineering & Applications, 2001 International Symposium on,
16-18 July 2001 Page(s):330 - 336
Digital Object Identifier 10.1109/IDEAS.2001.938101

AbstractPlus | Full Text: PDF(516 KB) IEEE CNF
Rights and Permissions

14. **An approach to functional dependencies in fuzzy databases**
Bosc, P.; Pivert, O.; Lietrd, L.;
Fuzzy Systems, 2000. FUZZ IEEE 2000. The Ninth IEEE International Conference on
Volume 2, 7-10 May 2000 Page(s):917 - 922 vol.2
Digital Object Identifier 10.1109/FUZZY.2000.839154

AbstractPlus | Full Text: PDF(244 KB) IEEE CNF
Rights and Permissions

15. **Maintenance of placement dependency in distributed multidatabase systems**
Hao Chen; Chengwen Liu;
Database Systems for Advanced Applications, 1999. Proceedings., 6th International Conference
19-21 April 1999 Page(s):339 - 346
Digital Object Identifier 10.1109/DASFAA.1999.765768

AbstractPlus | Full Text: PDF(212 KB) IEEE CNF
Rights and Permissions

16. **Data level inference detection in database systems**
Yip, R.W.; Levitt, E.N.;
Computer Security Foundations Workshop, 1998. Proceedings, 11th IEEE
9-11 June 1998 Page(s):179 - 189
Digital Object Identifier 10.1109/CSFW.1998.683168

AbstractPlus | Full Text: PDF(240 KB) IEEE CNF
Rights and Permissions

17. **Defeasible dependencies for relational database design**
Potter, W.D.;
System Sciences, 1997, Proceedings of the Thirtieth Hawaii International Conference on
Volume 5, 7-10 Jan. 1997 Page(s):434 - 441 vol.5
Digital Object Identifier 10.1109/HICSS.1997.663202

AbstractPlus | Full Text: PDF(704 KB) IEEE CNF
Rights and Permissions

18. **Recursive functions with implicit termination: a new approach illustrated by software-engi**
Linnemann, V.;
System Sciences, 1990., Proceedings of the Twenty-Third Annual Hawaii International Conferenc
Volume ii, 2-5 Jan. 1990 Page(s):318 - 327 vol.2
Digital Object Identifier 10.1109/HICSS.1990.205203

AbstractPlus | Full Text: PDF(708 KB) IEEE CNF
Rights and Permissions

19. **On semantic query optimization in deductive databases**
Lakshmanan, L.V.S.; Missaoui, R.;
Data Engineering, 1992. Proceedings, Eighth International Conference on
2-3 Feb. 1992 Page(s):368 - 375
Digital Object Identifier 10.1109/ICDE.1992.213173

AbstractPlus | Full Text: PDF(816 KB) IEEE CNF

Rights and Permissions

20. **Deduction graphs: an algorithm and applications**
Chao-Chih Yang;
Software Engineering, IEEE Transactions on
Volume 15, Issue 1, Jan. 1989 Page(s):60 - 67
Digital Object Identifier 10.1109/32.21726
AbstractPlus | Full Text: PDF(700 KB) IEEE JNL
Rights and Permissions

21. **Distributed query processing a multiple database system**
Chen, A.L.P.; Brill, D.; Templeton, M.; Yu, C.T.;
Selected Areas in Communications, IEEE Journal on
Volume 7, Issue 3, April 1989 Page(s):390 - 398
Digital Object Identifier 10.1109/49.16871
AbstractPlus | Full Text: PDF(864 KB) IEEE JNL
Rights and Permissions

22. **Reasoning about equations and functional dependencies on complex objects**
van Bommel, M.F.; Weddell, G.E.;
Knowledge and Data Engineering, IEEE Transactions on
Volume 6, Issue 3, June 1994 Page(s):455 - 469
Digital Object Identifier 10.1109/69.334857
AbstractPlus | Full Text: PDF(1188 KB) IEEE JNL
Rights and Permissions

23. **An object-oriented query evaluation scheme for logical databases in massively parallel en**
Lee, W.S.; Sheu, P.C.-Y.;
Knowledge and Data Engineering, IEEE Transactions on
Volume 6, Issue 1, Feb. 1994 Page(s):181 - 187
Digital Object Identifier 10.1109/69.273036
AbstractPlus | Full Text: PDF(704 KB) IEEE JNL
Rights and Permissions

24. **Using constraints for efficient query processing in nondeterministic databases**
Vadaparty, K.; Naqvi, S.;
Knowledge and Data Engineering, IEEE Transactions on
Volume 7, Issue 6, Dec. 1995 Page(s):850 - 864
Digital Object Identifier 10.1109/69.476493
AbstractPlus | References | Full Text: PDF(1712 KB) IEEE JNL
Rights and Permissions

25. **A real-time matching system for large fingerprint databases**
Ratha, N.K.; Karu, K.; Shaoyun Chen; Jain, A.K.;
Pattern Analysis and Machine Intelligence, IEEE Transactions on
Volume 18, Issue 8, Aug. 1996 Page(s):799 - 813
Digital Object Identifier 10.1109/34.531800
AbstractPlus | References | Full Text: PDF(2016 KB) IEEE JNL
Rights and Permissions

Results for "(((querying)<in>metadata) <and> ((database)<in>metadata))<and> ((depend..."
Your search matched **92** of **1685772** documents.
A maximum of **92** results are displayed, **25** to a page, sorted by **Relevance** in **Descending** order.

26. **Reasoning about uniqueness constraints in object relational databases**
Khizder, V.L.; Weddell, G.E.;
Knowledge and Data Engineering, IEEE Transactions on
Volume 15, Issue 5, Sept.-Oct. 2003 Page(s):1295 - 1306
Digital Object Identifier 10.1109/TKDE.2003.1232279
AbstractPlus | References | Full Text: PDF(759 KB) IEEE JNL
Rights and Permissions

27. **QPIAD: Query Processing over Incomplete Autonomous Databases**
Khatri, H.; Jianchun Fan; Yi Chen; Kambhampati, S.;
Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on
15-20 April 2007 Page(s):1430 - 1432
Digital Object Identifier 10.1109/ICDE.2007.369028
AbstractPlus | Full Text: PDF(95 KB) IEEE CNF
Rights and Permissions

28. **Hierarchical metadata driven view dependency in spatial data warehouses**
Yu, S.;
Database and Expert Systems Applications, 2005. Proceedings. Sixteenth International Worksho
22-26 Aug. 2005 Page(s):448 - 452
Digital Object Identifier 10.1109/DEXA.2005.105
AbstractPlus | Full Text: PDF(93 KB) IEEE CNF
Rights and Permissions

29. **Comprehending and visualizing software based on XML-representations and call graphs**
Hopfner, M.; Seipel, D.; von Gudenberg, J.W.;
Program Comprehension, 2003. 11th IEEE International Workshop on
10-11 May 2003 Page(s):290 - 291
Digital Object Identifier 10.1109/WPC.2003.1199218
AbstractPlus | Full Text: PDF(254 KB) IEEE CNF
Rights and Permissions

30. **Machining feature-based comparisons of mechanical parts**

Cicirello, V.; Regli, W.C.;
Shape Modeling and Applications, SMI 2001 International Conference on.
7-11 May 2001 Page(s):176 - 185
Digital Object Identifier 10.1109/SMA.2001.923388

AbstractPlus | Full Text: PDF(824 KB) IEEE CNF
Rights and Permissions

31. **Complements for data warehouses**
Laurent, D.; Lechtenborger, J.; Spyratos, N.; Vossen, G.;
Data Engineering, 1999. Proceedings., 15th International Conference on
23-26 March 1999 Page(s):490 - 499
Digital Object Identifier 10.1109/ICDE.1999.754965

AbstractPlus | Full Text: PDF(176 KB) IEEE CNF
Rights and Permissions

32. **FRAGTIQUE: an OO distribution design methodology**
Savonnet, M.; Terrasse, M.-N.; Yetongnon, K.;
Database Systems for Advanced Applications, 1999. Proceedings., 6th International Conference
19-21 April 1999 Page(s):283 - 290
Digital Object Identifier 10.1109/DASFAA.1999.765762

AbstractPlus | Full Text: PDF(152 KB) IEEE CNF
Rights and Permissions

33. **Location dependent data and its management in mobile databases**
Dunham, M.H.; Kumar, V.;
Database and Expert Systems Applications, 1998. Proceedings. Ninth International Workshop on
26-28 Aug. 1998 Page(s):414 - 419
Digital Object Identifier 10.1109/DEXA.1998.707433

AbstractPlus | Full Text: PDF(200 KB) IEEE CNF
Rights and Permissions

34. **Proceedings 14th International Conference on Data Engineering**
Data Engineering, 1998. Proceedings., 14th International Conference on
23-27 Feb. 1998
Digital Object Identifier 10.1109/ICDE.1998.655751

AbstractPlus | Full Text: PDF(568 KB) IEEE CNF
Rights and Permissions

35. **Knowledge discovery in relational database: extraction of n-ary relationships**
Soutou, C.;
Information Technology, 1997. BIWIT '97., Proceedings of the Third Basque International Worksh
2-4 July 1997 Page(s):46 - 53
Digital Object Identifier 10.1109/BIWIT.1997.614050

AbstractPlus | Full Text: PDF(748 KB) IEEE CNF
Rights and Permissions

36. **Analogical reasoning to answer null queries in fuzzy object-oriented data model**
Shyue-Liang Wang; Ta-Jung Huang;
Fuzzy Systems, 1997., Proceedings of the Sixth IEEE International Conference on
Volume 1, 1-5 July 1997 Page(s):31 - 36 vol.1
Digital Object Identifier 10.1109/FUZZY.1997.616340

AbstractPlus | Full Text: PDF(444 KB) IEEE CNF
Rights and Permissions

37. **Efficient processing of distributed set queries**
El-Sharkawi, M.E.; Kambayashi, Y.;
Databases, Parallel Architectures and Their Applications., PARBASE-90, International Conferenc
7-9 March 1990 Page(s):6 - 13
Digital Object Identifier 10.1109/PARBSE.1990.77111

AbstractPlus | Full Text: PDF(664 KB) IEEE CNF
Rights and Permissions

38. **pRETS: a parallel reverse-engineering tool set for the adaptation of sequential programs**
Gifford, B.; Harrison, W.;
Software Maintenance, 1990., Proceedings., Conference on
26-29 Nov. 1990 Page(s):344 - 346
Digital Object Identifier 10.1109/ICSM.1990.131385
AbstractPlus | Full Text: PDF(172 KB) IEEE CNF
Rights and Permissions

39. **Global knowledge about local dependencies in an integrated database environment**
Marinos, L.;
Tools for Artificial Intelligence, 1990., Proceedings of the 2nd International IEEE Conference on
6-9 Nov. 1990 Page(s):783 - 791
Digital Object Identifier 10.1109/TAI.1990.130438
AbstractPlus | Full Text: PDF(596 KB) IEEE CNF
Rights and Permissions

40. **A pattern based approach of integrating data and knowledge to support cooperative query**
Chu, W.W.; Chen, Q.;
Systems Integration, 1992. ICSI '92., Proceedings of the Second International Conference on
15-18 June 1992 Page(s):615 - 624
Digital Object Identifier 10.1109/ICSI.1992.217269
AbstractPlus | Full Text: PDF(704 KB) IEEE CNF
Rights and Permissions

41. **A study in software process data capture and analysis**
Wolf, A.L.; Rosenblum, D.S.;
Software Process, 1993. 'Continuous Software Process Improvement', Second International Con
25-26 Feb. 1993 Page(s):115 - 124
Digital Object Identifier 10.1109/SPCON.1993.236817
AbstractPlus | Full Text: PDF(792 KB) IEEE CNF
Rights and Permissions

42. **Approximate matchings in scientific databases**
Chatterjee, A.; Segev, A.;
System Sciences, 1994. Vol.III: Information Systems: Decision Support and Knowledge-Based Sy
Twenty-Seventh Hawaii International Conference on
Volume 3, 4-7 Jan. 1994 Page(s):448 - 457
Digital Object Identifier 10.1109/HICSS.1994.323327
AbstractPlus | Full Text: PDF(984 KB) IEEE CNF
Rights and Permissions

43. **Performance analysis of an object-oriented approach to parallel query evaluation**
Poola, T.R.; Lee, W.S.; Sheu, P.C.-Y.;
Computer Software and Applications Conference, 1994. COMPSAC 94. Proceedings., Eighteenth
9-11 Nov. 1994 Page(s):264 - 269
Digital Object Identifier 10.1109/CMPSAC.1994.342794
AbstractPlus | Full Text: PDF(464 KB) IEEE CNF
Rights and Permissions

44. **Statistical Models for Assessing the Individuality of Fingerprints**
Zhu, Y.; Dass, S. C.; Jain, A. K.;
Information Forensics and Security, IEEE Transactions on
Volume 2, Issue 3, Part 1, Sept. 2007 Page(s):391 - 401
Digital Object Identifier 10.1109/TIFS.2007.903846
AbstractPlus | Full Text: PDF(2214 KB) IEEE JNL

Rights and Permissions

45. **On the Correlation of Automatic Audio and Visual Segmentations of Music Videos**
Gillet, O.; Essid, S.; Richard, G.;
Circuits and Systems for Video Technology, IEEE Transactions on
Volume 17, Issue 3, March 2007 Page(s):347 - 355
Digital Object Identifier 10.1109/TCSVT.2007.890831
AbstractPlus | Full Text: PDF(549 KB) IEEE JNL
Rights and Permissions

46. **Converting Mathematical Models of Physiological Systems to Relational Database Scheme**
Yamamoto, William S.;
Biomedical Engineering, IEEE Transactions on
Volume BME-32, Issue 4, April 1985 Page(s):273 - 276
Digital Object Identifier 10.1109/TBME.1985.325448
AbstractPlus | Full Text: PDF(2014 KB) IEEE JNL
Rights and Permissions

47. **Kaleidoscope: a cooperative menu-guided query interface (SQL version)**
Cha, S.K.;
Knowledge and Data Engineering, IEEE Transactions on
Volume 3, Issue 1, March 1991 Page(s):42 - 47
Digital Object Identifier 10.1109/69.75887
AbstractPlus | Full Text: PDF(324 KB) IEEE JNL
Rights and Permissions

48. **On the "dimensionality curse" and the "self-similarity blessing"**
Korn, F.; Pagel, B.-U.; Faloutsos, C.;
Knowledge and Data Engineering, IEEE Transactions on
Volume 13, Issue 1, Jan.-Feb. 2001 Page(s):96 - 111
Digital Object Identifier 10.1109/69.908983
AbstractPlus | References | Full Text: PDF(1132 KB) IEEE JNL
Rights and Permissions

49. **Methods for Key-Based Conflicts of Data Organization in Information Integrated Systems**
Zhao, Ling-Qiang; Gu, Zhao-Peng; Hao, Zhong-Xiao;
Machine Learning and Cybernetics, 2007 International Conference on
Volume 2, 19-22 Aug. 2007 Page(s):781 - 788
Digital Object Identifier 10.1109/ICMLC.2007.4370249
AbstractPlus | Full Text: PDF(980 KB) IEEE CNF
Rights and Permissions

50. **Anomaly Detection in XML databases by means of Association Rules**
Bruno, Giulia; Garza, Paolo; Quintarelli, Elisa; Rossato, Rosalba;
Database and Expert Systems Applications, 2007. DEXA '07. 18th International Conference on
3-7 Sept. 2007 Page(s):387 - 391
Digital Object Identifier 10.1109/DEXA.2007.4312922
AbstractPlus | Full Text: PDF(138 KB) IEEE CNF
Rights and Permissions

Results for "(((querying)<in>metadata) <and> ((database)<in>metadata))<and> ((depend..."
Your search matched 92 of 1685772 documents.
A maximum of 92 results are displayed, 25 to a page, sorted by Relevance in Descending order.

Modify Search

(((querying)<in>metadata) <and> ((database)<in>metadata))<and> ((dependency) Search

Check to search only within this results set

Display Format: Citation Citation & Abstract

» Search Options

View Session History

New Search

IEEE/IET Books Educational Courses

IEEE/IET journals, transactions, letters, magazines, conference proceedings, and standards.

» Key

IEEE JNL	IEEE Journal or Magazine
IET JNL	IET Journal or Magazine
IEEE CNF	IEEE Conference Proceeding
IET CNF	IET Conference Proceeding
IEEE STD	IEEE Standard

view selected items Select All Deselect All

51. **A Method to Compress Schema-Based XML Metadata for Mobile Environments**
Fang, Jianjun; Martinez-Smith, Alfonso; Gandhi, Bhavan;
Multimedia and Expo, 2007 IEEE International Conference on
2-5 July 2007 Page(s):719 - 722
Digital Object Identifier 10.1109/ICME.2007.4284751
AbstractPlus | Full Text: PDF(236 KB) IEEE CNF
Rights and Permissions

52. **Fast Identification of Relational Constraint Violations**
Chandel, A.; Koudas, N.; Pu, K.Q.; Srivastava, D.;
Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on
15-20 April 2007 Page(s):776 - 785
Digital Object Identifier 10.1109/ICDE.2007.367923
AbstractPlus | Full Text: PDF(3391 KB) IEEE CNF
Rights and Permissions

53. **Visual Modeling of Behavioral Properties in the LVM for XML Using XSemantic Nets**
Rajugan, R.; Chang, E.; Feng, L.; Dillon, T.S.;
Database Engineering and Applications Symposium, 2006. IDEAS '06. 10th International
Dec. 2006 Page(s):209 - 219
Digital Object Identifier 10.1109/IDEAS.2006.51
AbstractPlus | Full Text: PDF(580 KB) IEEE CNF
Rights and Permissions

54. **A Fast Density-Based Clustering Algorithm for Large Databases**
Bing Liu;
Machine Learning and Cybernetics, 2006 International Conference on
Aug. 2006 Page(s):996 - 1000
Digital Object Identifier 10.1109/ICMLC.2006.258531
AbstractPlus | Full Text: PDF(291 KB) IEEE CNF
Rights and Permissions

55. **Mapping Relational Schemas to XML DTDs with Constraints**

Teng Lv; Ping Yan;
Computer and Computational Sciences, 2006. IMSCCS '06. First International Multi-Symposiums
Volume 2, 20-24 April 2006 Page(s):528 - 533
Digital Object Identifier 10.1109/IMSCCS.2006.237

AbstractPlus | Full Text: PDF(224 KB) IEEE CNF
Rights and Permissions

56. **Statistical models for assessing the individuality of fingerprints**
Dass, S.C.; Zhu, Y.; Jain, A.K.;
Automatic Identification Advanced Technologies, 2005. Fourth IEEE Workshop on
17-18 Oct. 2005 Page(s):3 - 9
Digital Object Identifier 10.1109/AUTOID.2005.46

AbstractPlus | Full Text: PDF(384 KB) IEEE CNF
Rights and Permissions

57. **Order-Sensitive XML Query Processing over Relational Sources: An Algebraic Approach**
Ling Wang; Song Wang; Murphy, B.; Rundensteiner, E.A.;
Database Engineering and Application Symposium, 2005. IDEAS 2005. 9th International
25-27 July 2005 Page(s):175 - 184
Digital Object Identifier 10.1109/IDEAS.2005.40

AbstractPlus | Full Text: PDF(336 KB) IEEE CNF
Rights and Permissions

58. **Detection and correction of conflicting source updates for view maintenance**
Chen, S.; Chen, J.; Zhang, X.; Rundensteiner, E.A.;
Data Engineering, 2004. Proceedings. 20th International Conference on
30 March-2 April 2004 Page(s):436 - 447
Digital Object Identifier 10.1109/ICDE.2004.1320017

AbstractPlus | Full Text: PDF(441 KB) IEEE CNF
Rights and Permissions

59. **Automated mining of granular database scheme**
Wong, S.K.M.; Wu, D.;
Fuzzy Systems, 2002. FUZZ-IEEE'02. Proceedings of the 2002 IEEE International Conference on
Volume 1, 12-17 May 2002 Page(s):690 - 694
Digital Object Identifier 10.1109/FUZZ.2002.1005076

AbstractPlus | Full Text: PDF(410 KB) IEEE CNF
Rights and Permissions

60. **An efficient algorithm for processing distributed queries using partition dependency**
Chen, H.; Liu, C.;
Parallel and Distributed Systems, 2000. Proceedings. Seventh International Conference on
4-7 July 2000 Page(s):339 - 346
Digital Object Identifier 10.1109/ICPADS.2000.857716

AbstractPlus | Full Text: PDF(732 KB) IEEE CNF
Rights and Permissions

61. **Deflating the dimensionality curse using multiple fractal dimensions**
Pagel, B.-U.; Korn, F.; Faloutsos, C.;
Data Engineering, 2000. Proceedings. 16th International Conference on
29 Feb.-3 March 2000 Page(s):589 - 598
Digital Object Identifier 10.1109/ICDE.2000.839457

AbstractPlus | Full Text: PDF(288 KB) IEEE CNF
Rights and Permissions

62. **Query folding with inclusion dependencies**
Gryz, J.;
Data Engineering, 1998. Proceedings., 14th International Conference on
23-27 Feb. 1998 Page(s):126 - 133

Digital Object Identifier 10.1109/ICDE.1998.655768

AbstractPlus | Full Text: PDF(108 KB) IEEE CNF
Rights and Permissions

63. **Object Wrapper: an object-oriented interface for relational databases**
Bergamaschi, S.; Garuti, A.; Sartori, C.; Venuta, A.;
EUROMICRO 97. 'New Frontiers of Information Technology'., Proceedings of the 23rd EUROMIC
1-4 Sept. 1997 Page(s):41 - 46
Digital Object Identifier 10.1109/EURMIC.1997.617214

AbstractPlus | Full Text: PDF(448 KB) IEEE CNF
Rights and Permissions

64. **Operational information system in a power plant**
Ordieres Mere, J.; Ortega, F.; Bello, A.; Menendez, C.; Vallina, V.;
Systems, Man, and Cybernetics, 1997. 'Computational Cybernetics and Simulation'., 1997 IEEE I
Volume 4, 12-15 Oct. 1997 Page(s):3285 - 3288 vol.4
Digital Object Identifier 10.1109/ICSMC.1997.633121

AbstractPlus | Full Text: PDF(296 KB) IEEE CNF
Rights and Permissions

65. **Statistical dependencies**
Cotofrei, P.; Luchian, H.;
Scientific and Statistical Database Systems, 1996. Proceedings., Eighth International Conference
18-20 June 1996 Page(s):32 - 41
Digital Object Identifier 10.1109/SSDM.1996.505913

AbstractPlus | Full Text: PDF(676 KB) IEEE CNF
Rights and Permissions

66. **Record subtyping in flexible relations by means of attribute dependencies**
Kalus, C.; Dadam, P.;
Data Engineering, 1995. Proceedings of the Eleventh International Conference on
6-10 March 1995 Page(s):383 - 390
Digital Object Identifier 10.1109/ICDE.1995.380359

AbstractPlus | Full Text: PDF(672 KB) IEEE CNF
Rights and Permissions

67. **An object-based query evaluation scheme for deductive databases in massively parallel co**
Lee, W.S.; Sheu, P.C.;
Data Engineering, 1989. Proceedings, Fifth International Conference on
6-10 Feb. 1989 Page(s):497 - 504
Digital Object Identifier 10.1109/ICDE.1989.47255

AbstractPlus | Full Text: PDF(800 KB) IEEE CNF
Rights and Permissions

68. **Implicit joins in the structural data model**
Lee, B.S.; Litwin, W.; Wiederhold, G.;
Computer Software and Applications Conference, 1991. COMPSAC '91., Proceedings of the Fifte
11-13 Sept. 1991 Page(s):357 - 364
Digital Object Identifier 10.1109/CMPSAC.1991.170203

AbstractPlus | Full Text: PDF(616 KB) IEEE CNF
Rights and Permissions

69. **Visual strategies for querying databases**
Batini, C.; Catarci, T.; Costabile, M.F.; Levialdi, S.;
Visual Languages, 1991., Proceedings, 1991 IEEE Workshop on
8-11 Oct. 1991 Page(s):183 - 189
Digital Object Identifier 10.1109/WVL.1991.238834

AbstractPlus | Full Text: PDF(724 KB) IEEE CNF

Rights and Permissions

70. **The application of deductive databases to inter-module code analysis**
Dietrich, S.W.; Calliss, F.W.;
Software Maintenance, 1991., Proceedings, Conference on
15-17 Oct. 1991 Page(s):120 - 128
Digital Object Identifier 10.1109/ICSM.1991.160318
AbstractPlus | Full Text: PDF(624 KB) IEEE CNF
Rights and Permissions

71. **Processing of multiple processing queries in distributed databases**
Lu, A.Y.; Sheu, P.C.-Y.;
Data Engineering, 1991, Proceedings, Seventh International Conference on
8-12 April 1991 Page(s):42 - 49
Digital Object Identifier 10.1109/ICDE.1991.131451
AbstractPlus | Full Text: PDF(584 KB) IEEE CNF
Rights and Permissions

72. **MULS: A General Framework of Providing Multilevel Service Quality in Sequential Data Bro**
Hung, Hao-Ping; Chen, Ming-Syan;
Transactions on Knowledge and Data Engineering
Volume 19, Issue 10, Oct. 2007 Page(s):1433 - 1447
Digital Object Identifier 10.1109/TKDE.2007.1072
AbstractPlus | Full Text: PDF(3021 KB) IEEE JNL
Rights and Permissions

73. **A compensation-based approach for view maintenance in distributed environments**
Chen, S.; Zhang, X.; Rundensteiner, E.A.;
Knowledge and Data Engineering, IEEE Transactions on
Volume 18, Issue 8, Aug. 2006 Page(s):1068 - 1081
Digital Object Identifier 10.1109/TKDE.2006.117
AbstractPlus | Full Text: PDF(1848 KB) IEEE JNL
Rights and Permissions

74. **Optimization of parallel execution for multi-join queries**
Ming-Syan Chen; Yu, P.S.; Kun-Lung Wu;
Knowledge and Data Engineering, IEEE Transactions on
Volume 8, Issue 3, June 1996 Page(s):416 - 428
Digital Object Identifier 10.1109/69.506709
AbstractPlus | References | Full Text: PDF(1352 KB) IEEE JNL
Rights and Permissions

75. **An efficient indexing scheme for image storage and recognition**
Al-Mouhamed, M.;
Industrial Electronics, IEEE Transactions on
Volume 46, Issue 2, April 1999 Page(s):429 - 439
Digital Object Identifier 10.1109/41.753782
AbstractPlus | References | Full Text: PDF(288 KB) IEEE JNL
Rights and Permissions

Results for "(((querying)<in>metadata) <and> ((database)<in>metadata))<and> ((depend..."
Your search matched 92 of 1685772 documents.
A maximum of 92 results are displayed, 25 to a page, sorted by Relevance in Descending order.

76. **Rewriting queries using views**
Flesca, S.; Greco, S.;
Knowledge and Data Engineering, IEEE Transactions on
Volume 13, Issue 6, Nov.-Dec. 2001 Page(s):980 - 995
Digital Object Identifier 10.1109/69.971191
AbstractPlus | References | Full Text: PDF(386 KB) IEEE JNL
Rights and Permissions

77. **Performance Characterization of BLAST for the Grid**
Afgan, Enis; Bangalore, Purushotham;
Bioinformatics and Bioengineering, 2007. BIBE 2007. Proceedings of the 7th IEEE International C
14-17 Oct. 2007 Page(s):1394 - 1398
Digital Object Identifier 10.1109/BIBE.2007.4375754
AbstractPlus | Full Text: PDF(635 KB) IEEE CNF
Rights and Permissions

78. **TBF: An Efficient Data Architecture for Metadata Server in the Object-based Storage Netwo**
Yu Hua; Feng, D.; Bin Xiao;
Networks, 2006. ICON '06. 14th IEEE International Conference on
Volume 1, Sept. 2006 Page(s):1 - 6
Digital Object Identifier 10.1109/ICON.2006.302603
AbstractPlus | Full Text: PDF(839 KB) IEEE CNF
Rights and Permissions

79. **Relational Model of Data over Domains with Similarities: An Extension for Similarity Queri**
Belohlavek, R.; Vychodil, V.;
Information Reuse and Integration, 2006 IEEE International Conference on
Sept. 2006 Page(s):207 - 213
Digital Object Identifier 10.1109/IRI.2006.252414
AbstractPlus | Full Text: PDF(158 KB) IEEE CNF
Rights and Permissions

80. **Co-existence of transaction and non transaction-managed activity in a persistent object st**

Henskens, F.A.; Ashton, M.G.;
Systems Engineering, 2005. ICSEng 2005. 18th International Conference on
16-18 Aug. 2005 Page(s):81 - 86
Digital Object Identifier 10.1109/ICSENG.2005.26

AbstractPlus | Full Text: PDF(96 KB) IEEE CNF
Rights and Permissions

81. **A simple extension of queriable compression for XML data**
Maita, T.; Sakamoto, H.;
Active Media Technology, 2005. (AMT 2005). Proceedings of the 2005 International Conference c
19-21 May 2005 Page(s):91 - 95
Digital Object Identifier 10.1109/AMT.2005.1505276

AbstractPlus | Full Text: PDF(255 KB) IEEE CNF
Rights and Permissions

82. **Zigzag: a new algorithm for mining large inclusion dependencies in databases**
De Marchi, F.; Petit, J.-M.;
Data Mining, 2003. ICDM 2003. Third IEEE International Conference on
19-22 Nov. 2003 Page(s):27 - 34

AbstractPlus | Full Text: PDF(345 KB) IEEE CNF
Rights and Permissions

83. **CataLogger: a framework for building asset management solutions**
Hamilton, G.;
Computer Software and Applications Conference, 1999. COMPSAC '99. Proceedings. The Twent
27-29 Oct. 1999 Page(s):328 - 329
Digital Object Identifier 10.1109/CMPSAC.1999.812731

AbstractPlus | Full Text: PDF(148 KB) IEEE CNF
Rights and Permissions

84. **A model for deferred view maintenance**
Alhajj, R.; Elnagar, A.;
Database Engineering and Applications, 1999. IDEAS '99. International Symposium Proceedings
2-4 Aug. 1999 Page(s):188 - 196
Digital Object Identifier 10.1109/IDEAS.1999.787267

AbstractPlus | Full Text: PDF(580 KB) IEEE CNF
Rights and Permissions

85. **Proceedings. Fifth International Workshop on Temporal Representation and Reasoning (Cı**
Temporal Representation and Reasoning, 1998. Proceedings. Fifth International Workshop on
16-17 May 1998
Digital Object Identifier 10.1109/TIME.1998.674123

AbstractPlus | Full Text: PDF(224 KB) IEEE CNF
Rights and Permissions

86. **TRICS: transformation system of relations into conceptual structure**
Kajiyama, T.; Ito, H.; Fukumura, T.;
Systems, Man, and Cybernetics, 1997. 'Computational Cybernetics and Simulation'., 1997 IEEE I
Volume 2, 12-15 Oct. 1997 Page(s):1390 - 1395 vol.2
Digital Object Identifier 10.1109/ICSMC.1997.638168

AbstractPlus | Full Text: PDF(568 KB) IEEE CNF
Rights and Permissions

87. **Using RSDM to mine socio-economic indicators**
Covadonga Fernandez-Baizan, M.; Menasalvas Ruiz, E.; Pena Sanchez, J.; Milian, S.;
Systems, Man, and Cybernetics, 1997. 'Computational Cybernetics and Simulation'., 1997 IEEE I
Volume 2, 12-15 Oct. 1997 Page(s):1385 - 1389 vol.2
Digital Object Identifier 10.1109/ICSMC.1997.638167

AbstractPlus | Full Text: PDF(312 KB) IEEE CNF
Rights and Permissions

88. **Discovering conceptual structure of relations**
Ito, H.; Kajiyama, T.; Fukumura, T.;
Systems, Man, and Cybernetics, 1996., IEEE International Conference on
Volume 1, 14-17 Oct. 1996 Page(s):591 - 596 vol.1
Digital Object Identifier 10.1109/ICSMC.1996.569858
AbstractPlus | Full Text: PDF(492 KB) IEEE CNF
Rights and Permissions

89. **Lassy-a system for analyzing grammatical dependencies in Latin**
Gartner, H.; Gogolla, M.;
Database and Expert Systems Applications, 1996. Proceedings., Seventh International Workshop
9-10 Sept. 1996 Page(s):275 - 284
Digital Object Identifier 10.1109/DEXA.1996.558318
AbstractPlus | Full Text: PDF(800 KB) IEEE CNF
Rights and Permissions

90. **Consistent view removal in transparent schema evolution systems**
Crestana-Taube, V.; Rundensteiner, E.A.;
Research Issues in Data Engineering, 1996. Interoperability of Nontraditional Database Systems. Workshop on
26-27 Feb. 1996 Page(s):138 - 147
Digital Object Identifier 10.1109/RIDE.1996.492251
AbstractPlus | Full Text: PDF(896 KB) IEEE CNF
Rights and Permissions

91. **Algebraic optimization for nested relations**
Jan, Y.;
System Sciences, 1990., Proceedings of the Twenty-Third Annual Hawaii International Conferenc
Volume ii, 2-5 Jan. 1990 Page(s):278 - 287 vol.2
Digital Object Identifier 10.1109/HICSS.1990.205199
AbstractPlus | Full Text: PDF(508 KB) IEEE CNF
Rights and Permissions

92. **Semantic change computation optimization in active databases**
Urpi, T.; Olive, A.;
Research Issues in Data Engineering, 1994. Active Database Systems, Proceedings Fourth Inter
14-15 Feb. 1994 Page(s):19 - 27
Digital Object Identifier 10.1109/RIDE.1994.282857
AbstractPlus | Full Text: PDF(548 KB) IEEE CNF
Rights and Permissions

Search

Full Text
Concept
Document ID
Recent Disclosures

Other

Prior Art Home
Support
Logout

Displaying records #1 through 5 out of 5

Result # 1 Relevance:

DNS Security Introduction and Requirements (RFC4033)

2005-03-01 IPCOM000111193D En

The Domain Name System Security Extensions (DNSSEC) add data origin authenticatio integrity to the Domain Name System. This document introduces these extensions and capabilities and limitations. This document also discusses the services that the ...

Result # 2 Relevance:

BACKBORD: Beyond Retrival by Reformulation

1988-03-31 IPCOM000151988D En

Unclassified SECURITY CLASSIFICATION OF THIS PAGE REPORT DOCUMENTATION PAG SECURITY CLASSIFICATION Unclassified 2a. SECURITY CLASSIFICATION AUTHORITY 1 RESTRICTIVE MARKINGS 3 . DISTRIBUTION AVAILABILITY OF REPORT This document i for ...

Result # 3 Relevance:

BACKBORD Beyond Retrieval by Reformulation

1988-12-31 IPCOM000128683D En

Figure 4-1: A BACKBORD screen for inspecting the immediate children of 6 Notes-On-Is 2: A BACKBORD screen for browsing the knowledge base graphically 7 Figure 4-3: A BA screen for creating a knowledge acquisition note 9 Figure 4-4: A BACKBORD screen for

---

Result # 4 Relevance:

DESIGN OVERVIEW OF THE NAIL! SYSTEM

1899-12-30 IPCOM000150521D En

DESIGN OVERVIEW OF THE NAIL! SYSTEMt Katherine Morris Jeffrey D. UllmanAllen Var anford University Stanford, CA 94305, USA ABSTRACT: We describe the design decisior the NAIL! (not an- other implementation of logic!) system, am advanced form of DBMS

Result # 5 Relevance:

OPTIMIZING DATALOG PROGRAMS

1899-12-30 IPCOM000150529D En

OPTIMIZING DATALOG PROGRAMS Yehoshua Sa.givtStanf01.d Uni c-ersity Datnlog pro Prolog programs without functioll symbols, are considered. It is assunlecl that a variabl the head uf a rule must a.lso appear in the body of the rule. The input of a ...

Displaying page 1 of 1 << FIRST | < BACK | NEXT > | LAST >>

Search query: (recursive query) and dependency

New search | Modify this search | Search within current results

dogpile®

Web | Images | Audio | Video | News | Yellow Pages | White Pages

recursive query dependency procedure Go Fetch! Advanced Search Preferences

Now Searching: Google YAHOO! SEARCH Live Search Ask Learn More

Web Search Results for "recursive query dependency procedur..."

Search Filter: Moderate

Now Searching Google YAHOO! SEARCH Live Search Ask And More...

1 - 20 of 50 from All Search Engines (About Results)

1 | 2 | 3 Next >

1. On the decidability and finite controllability of query processing ...
   The kinds of ICs that we consider are functional dependencies (in particular key ... **Recursive query** plans for data integration. J. of Logic Programming, ...
   portal.acm.org/citation.cfm?id=1142351.1142404
   [Found on Google]

2. SYSTEM AND METHOD FOR RECURSIVE PATH ANALYSIS OF DBMS PROCEDURES - Patent 20010049...
   ... and then for each **dependency** found, doing the **query** recursively until all basic ... for the **procedure** that is being debugged and all its dependent procedures. ...
   www.freepatentsonline.com/20010049682.html
   [Found on Yahoo! Search, Ask.com]

3. tcpdump - Linux Command - Unix Command
   Linux / Unix Command Library: tcpdump. Learn about its synopsis, description, options, and examples.
   linux.about.com/library/cmd/blcmdl8_tcpdump.htm
   [Found on About]

4. SYSTEM AND METHOD FOR RECURSIVE PATH ANALYSIS OF DBMS PROCEDURES ...
   Using the **recursive procedure** in Step 1 to determine the dependencies of implementations of object oriented code objects, based on a **query** against the ...
   www.freepatentsonline.com/EP1208459.html
   [Found on Google]

5. Oracle Dependency Management
   ... a **query** that references tables or other views; a procedure's body can ... **procedure** is invalidated if any timestamps of remotely referenced procedures do ...
   www.csee.umbc.edu/help/oracle8/server.815/a67781/c...
   [Found on Yahoo! Search]

6. Citations: A Generalization of the Differential Approach to ...
   Balbin, I. and Ramamohanarao, K. A Generalization of the Differential Approach to **Recursive Query** Relations. Journal of Logic Programming to ...
   citeseer.comp.nus.edu.sg/context/158889/0
   [Found on Ask.com]

7. 6 Dependencies Among Schema Objects
   ... time stamp checking **dependency** model, whenever a **procedure** is compiled or ... all other local procedures that depend on the remote **procedure** with the new time ...
   www.stanford.edu/dept/itss/docs/oracle/10g/server....
   [Found on Yahoo! Search]

Are you looking for?
No Suggestions Found.

8. perl58delta - Linux Command - Unix Command
Linux / Unix Command Library: perl58delta. Learn about its synopsis, description, options, and examples.
linux.about.com/library/cmd/blcmdl1_perl58delta.ht...
[Found on About]

9. Query Reformulation in the Presence of Functional Dependencies ...
We first analyse the unification **procedure** of the MiniCon algorithm that plays ..... DUSCHKA, O.M., GENESERETH, M.R., LEVY, A.Y., **Recursive Query** Plans for ...
www.cs.qub.ac.uk/~J.Hong/iiwas03.doc [Found on Google]

10. Citations: An amateur's introduction to recursive query-processing ...
F. Bancilhon, and R. Ramakrishnan. An amateur's introduction to **recursive** query-processing strategies. In ACM SIGMOD, pages 16-52, 1986.
citeseer.ist.psu.edu/context/37532/0 [Found on Ask.com]

11. LNCS 2890 - Implication of Functional Dependencies for Recursive ...
method than computing the **recursive query** and after that checking whether the ... A proof **procedure** for data dependencies. J. ACM,. 31:718□741, 1984. ...
www.springerlink.com/index/0duykgprd9jx1v0n.pdf
[Found on Google]

12. Dependencies Among Schema Objects
... time stamp checking **dependency** model, whenever a **procedure** is compiled or ... all other local procedures that depend on the remote **procedure** with the new time ...
download-west.oracle.com/docs/cd/B10501_01/server....
[Found on Yahoo! Search]

13. Citations: Evaluation of Recursive Queries Using Join Indices - ...
Optimization Strategies for Parallel Linear **Recursive** Query.. ... techniques and operators for **recursive query** processing, e.g. in [Agr87] ...
citeseer.ist.psu.edu/context/74338/0 [Found on Ask.com]

14. Resolving circular dependency between view and procedure
Resolving circular **dependency** between view and **procedure** ... **Query** on dblink returning ORA-12545 error ...
whatis.techtarget.com/expert/KnowledgebaseAnswer/0...
[Found on Ask.com]

15. Chaotic fixpoint iteration guided by dynamic dependency
a side-effect modifies the dynamic **dependency** information. **Procedure** .... **recursive query** evaluation. J. Logic Programming 1987, 4, pp. 259-262. ...
www.springerlink.com/index/7k7u84362581351t2.pdf
[Found on Google]

16. Query Rewriting Using Views in the Presence of Inclusion Dependencies
dependency holds: procedure(procedure_name, patient_name) event(event_name, patient_name) ... A. Y. Levy, Recursive. **Query** Plans for Data Integration. ...
delivery.acm.org/10.1145/960000/956729/p134-bai.pd...
[Found on Yahoo! Search]

17. Resolving circular dependency between view and procedure
Resolving circular **dependency** between view and **procedure** ... 00604: error occurred at **recursive** level 1" and "ORA-01000: maximum open cursors ...
searchoracle.techtarget.com/expert/KnowledgebaseAn...
[Found on Ask.com]

18. Capturing both types and integrity constraints in data integration
DTD-conformance: recursive, nondeterministic; integrity constraints: validation .... scheduling: given a fixed **query dependency** graph, a heuristic based on ...
homepages.inf.ed.ac.uk/wenfei/talk/sigmod03.ppt
[Found on Google]

19. dbTalk Databases Forums - View Single Post - Using variables in ...
On Fri, 22 Jul 2005 01:06:01 -0700, Dunner wrote: Hi Hugo, What I am trying to do is write a **recursive query** to find the dependencies of a list of...
www.dbtalk.net/578032-post4.html [Found on Ask.com]

20. Divide-and-query and Subterm Dependency Tracking in the Mercury Declarative Debugg...
procedures, even if they represent di erent modes of the ... uni cation) in that **procedure** makes that variable ground. ... one of the two **recursive** calls, ...
www.cs.mu.oz.au/research/mercury/information/paper...
[Found on Yahoo! Search]

1 | 2 | 3 Next >

Oracle8*i* Concepts
Release 8.1.5
A67781-01

21
Oracle Dependency Management

Whoever you are--I have always depended on the kindness of strangers.

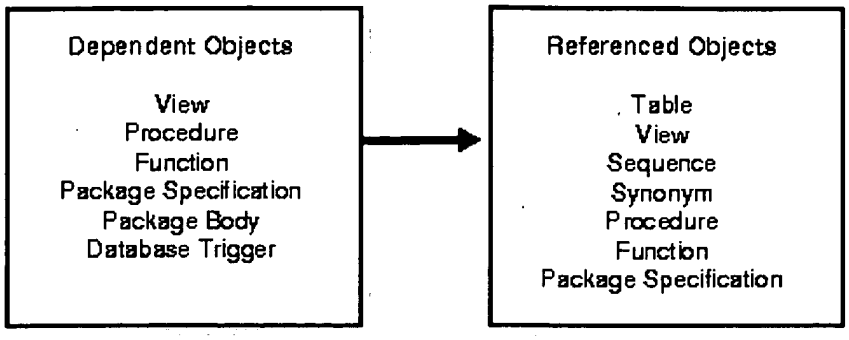
Tennessee Williams: *A Streetcar Named Desire*

The definitions of some objects, including views and procedures, reference other objects, such as tables. As a result, the objects being defined are dependent on the objects referenced in their definitions. This chapter discusses the dependencies among schema objects and how Oracle automatically tracks and manages these dependencies. It includes:

An Introduction to Dependency Issues

Some types of schema objects can reference other objects as part of their definition. For example, a view is defined by a query that references tables or other views; a procedure's body can include SQL statements that reference other objects of a database. An object that references another object as part of its definition is called a *dependent* object, while the object being referenced is a *referenced* object. Figure 21-1 illustrates the different types of dependent and referenced objects.

Figure 21-1 Types of Possible Dependent and Referenced Schema Objects



If you alter the definition of a referenced object, dependent objects may or may not continue to function without error, depending on the type of alteration. For example, if you drop a table, no view based on the dropped table can be used.

Oracle automatically records dependencies among objects to alleviate the complex job of dependency management for the database administrator and users. For example, if you alter a table on which several stored procedures depend, Oracle automatically recompiles the dependent procedures the next time the procedures are referenced (executed or compiled against).

To manage dependencies among schema objects, all of the schema objects in a database have a status:

VALID The schema object has been compiled and can be immediately used when referenced.
INVALID The schema object must be compiled before it can be used.

- In the case of procedures, functions, and packages, this means compiling the schema object.
- In the case of views, this means that the view must be reparsed, using the current definition in the data dictionary.

Only dependent objects can be invalid; tables, sequences, and synonyms are always valid.

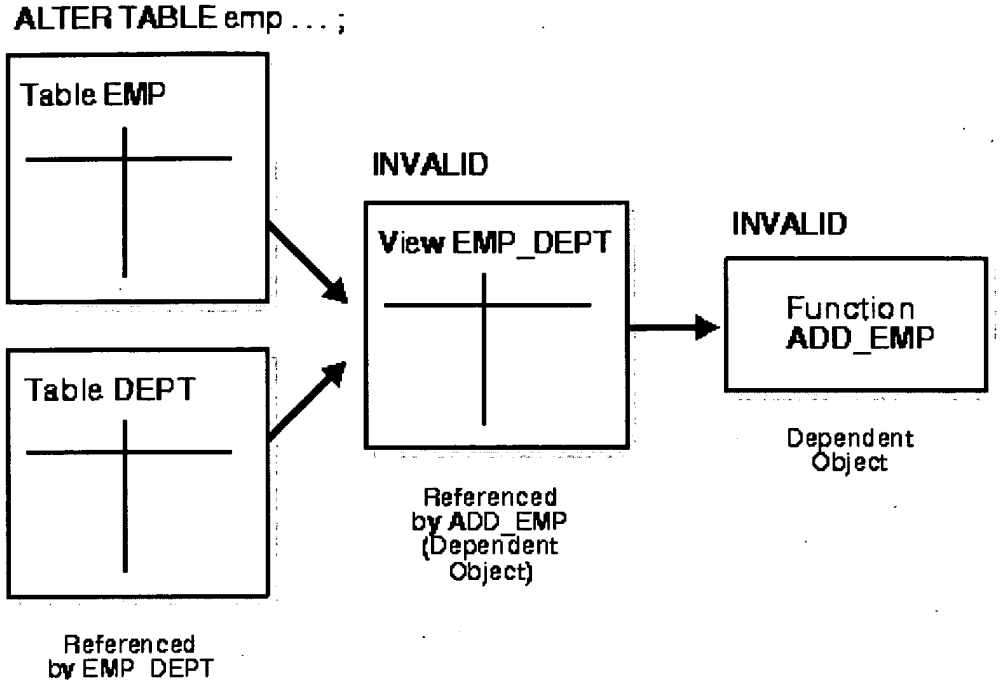
If a view, procedure, function, or package is invalid, Oracle may have attempted to compile it, but errors relating to the object occurred. For example, when compiling a view, one of its base tables might not exist, or the correct privileges for the base table might not be present. When compiling a package, there might be a PL/SQL or SQL syntax error, or the correct privileges for a referenced object might not be present. Schema objects with such problems remain invalid.

Oracle automatically tracks specific changes in the database and records the appropriate status for related objects in the data dictionary.

Status recording is a recursive process; any change in the status of a referenced object not only changes the status for directly dependent objects, but also for indirectly dependent objects.

For example, consider a stored procedure that directly references a view. In effect, the stored procedure indirectly references the base table(s) of that view. Therefore, if you alter a base table, the view is invalidated, which then invalidates the stored procedure. Figure 21-2 illustrates this.

Figure 21-2 Indirect Dependencies



Resolving Schema Object Dependencies

When a schema object is referenced (directly in a SQL statement or indirectly via a reference to a dependent object), Oracle checks the status of the object explicitly specified in the SQL statement and any referenced objects, as necessary. Oracle's action depends on the status of the objects that are directly and indirectly referenced in a SQL statement:

- If every referenced object is valid, Oracle executes the SQL statement immediately without any additional work.
- If any referenced view or procedure (including a function or package) is invalid, Oracle automatically attempts to compile the object.
    - If all invalid referenced objects can be compiled successfully, they are compiled and Oracle executes the SQL statement.
    - If an invalid object cannot be compiled successfully, it remains invalid and Oracle returns an error and rolls back the transaction containing the SQL statement.

Note:

Oracle attempts to recompile an invalid object dynamically only if it has not been replaced since it was detected as invalid. This optimization eliminates unnecessary recompilations.

Compiling Views and PL/SQL Program Units

A view or PL/SQL program unit can be compiled and made valid if the following conditions are satisfied:

- The definition of the view or program unit must be correct; all of the SQL and PL/SQL statements must be proper constructs.
- All referenced objects must be present and of the expected structure. For example, if the defining query of a view includes a column, the column must be present in the base table.
- The **owner** of the view or program unit must have the necessary privileges for the referenced objects. For example, if a SQL statement in a procedure inserts a row into a table, the owner of the procedure must have the INSERT privilege for the referenced table.

Views and Base Tables

A view depends on the base tables (or views) referenced in its defining query. If the defining query of a view is not explicit about which columns are referenced, for example, `SELECT * FROM table`, the defining query is expanded when stored in the data dictionary to include all columns in the referenced base table at that time.

If a base table (or view) of a view is altered, renamed, or dropped, the view is invalidated, but its definition remains in the data dictionary along with the privileges, synonyms, other objects, and other views that reference the invalid view.

An attempt to use an invalid view automatically causes Oracle to recompile the view dynamically. After replacing the view, the view might be valid or invalid, depending on the following condition:

- All base tables referenced by the defining query of a view must exist. If a base table of a view is renamed or dropped, the view is invalidated and cannot be used. References to invalid views cause the referencing statement to fail. The view can be compiled only if the base table is renamed to its original name or the base table is recreated.
- If a base table is altered or re-created with the same columns, but the datatype of one or more columns in the base table is changed, any dependent view can be recompiled successfully.
- If a base table of a view is altered or re-created with at least the same set of columns, the view can be validated. The view cannot be validated if the base table is re-created with new columns and the view references columns no longer contained in the re-created table. The latter point is especially relevant in the case of views defined with a `SELECT * FROM table` query, because the defining query is expanded at view creation time and permanently stored in the data dictionary.

Program Units and Referenced Objects

Oracle automatically invalidates a program unit when the definition of a referenced object is altered. For example, assume that a standalone procedure includes several statements that reference a table, a view, another standalone procedure, and a public package procedure. In that case, the following conditions

hold:

- If the referenced table is altered, the dependent procedure is invalidated.
- If the base table of the referenced view is altered, the view and the dependent procedure are invalidated.
- If the referenced standalone procedure is replaced, the dependent procedure is invalidated.
- If the *body* of the referenced package is replaced, the dependent procedure is not affected. However, if the *specification* of the referenced package is replaced, the dependent procedure is invalidated.

This last case reveals a mechanism for minimizing dependencies among procedures and referenced objects by using packages.

Session State and Referenced Packages

Each session that references a package construct has its own instance of that package, including a persistent state of any public and private variables, cursors, and constants. All of a session's package instantiations (including state) can be lost if any of the session's instantiated packages (specification or body) are subsequently invalidated and recompiled.

Security Authorizations

Oracle notices when a DML object or system privilege is granted to or revoked from a user or PUBLIC and automatically invalidates all the owner's dependent objects. Oracle invalidates the dependent objects to verify that an owner of a dependent object continues to have the necessary privileges for all referenced objects. Internally, Oracle notes that such objects do not have to be "recompiled"; only security authorizations need to be validated, not the structure of any objects. This optimization eliminates unnecessary recompilations and prevents the need to change a dependent object's timestamp.

Additional Information:

For information on forcing the recompilation of an invalid view or program unit, see the *Oracle8i Application Developer's Guide - Fundamentals*.

Function-Based Index Dependencies

Function-based indexes depend on functions used in the expression that defines the index. (See "Function-Based Indexes".) If such a function--a PL/SQL function or package function--is changed, the index is marked as disabled.

This section discusses requirements for function-based indexes and what happens when a function is changed in any manner, such as when it is dropped or privileges to use it are revoked.

Requirements

To create a function-based index:

- The following initialization parameters must be defined:
  - QUERY_REWRITE_INTEGRITY must be set to TRUSTED
  - QUERY_REWRITE_ENABLED must be set to TRUE
  - COMPATIBLE must set to 8.1.0.0.0 or a greater value
- The user must be granted CREATE INDEX and QUERY REWRITE, or CREATE ANY INDEX and GLOBAL QUERY REWRITE.

To use a function-based index:

- The table must be analyzed after the index is created.
- The query must be guaranteed not to need any NULL values from the indexed expression, since NULL values are not stored in indexes.

The following sections describe additional requirements.

DETERMINISTIC Functions

Any user-written function used in a function-based index must have been declared with the DETERMINISTIC keyword to indicate that the function will always return the same output return value for any given set of input argument values, now and in the future. See "DETERMINISTIC Functions" for more information.

Privileges on the Defining Function

The index owner needs the EXECUTE privilege on the function used to define a function-based index. If the EXECUTE privilege is revoked, Oracle marks the index DISABLED. The index owner does not need the EXECUTE WITH GRANT OPTION privilege on this function to grant SELECT privileges on the underlying table.

Resolving Dependencies of Function-Based Indexes

A function-based index depends on any function that it is using. If the function or the specification of a package containing the function is redefined (or if the index owner's EXECUTE privilege is revoked), the index is marked as DISABLED.

- Queries on a DISABLED index fail if the optimizer chooses to use the index.
- DML operations on a DISABLED index fail unless the index is also marked UNUSABLE and the initialization parameter SKIP_UNUSABLE_INDEXES is set to true.

To re-enable the index after a change to the function, use the ALTER INDEX ... ENABLE statement.

Dependency Management and Nonexistent Schema Objects

When a dependent object is created, Oracle attempts to resolve all references by first searching in the current schema. If a referenced object is not found in the current schema, Oracle attempts to resolve the reference by searching for a private synonym in the same schema. If a private synonym is not found, Oracle moves on, looking for a public synonym. If a public synonym is not found, Oracle searches for a schema name that matches the first portion of the object name. If a matching schema name is found, Oracle attempts to find the object in that schema. If no schema is found, an error is returned.

Because of how Oracle resolves references, it is possible for an object to depend on the *nonexistence* of other objects. This occurs when the dependent object uses a reference that would be interpreted differently were another object present. For example, assume the following:

- At the current point in time, the COMPANY schema contains a table named EMP.
- A PUBLIC synonym named EMP is created for COMPANY.EMP and the SELECT privilege for COMPANY.EMP is granted to the PUBLIC role.
- The JWARD schema does not contain a table or private synonym named EMP.
- The user JWARD creates a view in his schema with the following statement:

```
CREATE VIEW dept_salaries AS
   SELECT deptno, MIN(sal), AVG(sal), MAX(sal) FROM emp
   GROUP BY deptno
   ORDER BY deptno;
```

When JWARD creates the DEPT_SALARIES view, the reference to EMP is resolved by first looking for JWARD.EMP as a table, view, or private synonym, none of which is found, and then as a public synonym named EMP, which is found. As a result, Oracle notes that JWARD.DEPT_SALARIES depends on the nonexistence of JWARD.EMP and on the existence of PUBLIC.EMP.

Now assume that JWARD decides to create a new view named EMP in his schema using the following statement:

```
CREATE VIEW emp AS
   SELECT empno, ename, mgr, deptno
   FROM company.emp;
```

Notice that JWARD.EMP does not have the same structure as COMPANY.EMP.

As it attempts to resolve references in object definitions, Oracle internally makes note of dependencies that the new dependent object has on "nonexistent" objects--schema objects that, if they existed, would change the interpretation of the object's definition. Such dependencies must be noted in case a nonexistent object is later created. If a nonexistent object is created, all dependent objects must be invalidated so that dependent objects can be recompiled and verified and all dependent function-based indexes must be marked unusable.

Therefore, in the example above, as JWARD.EMP is created, JWARD.DEPT_SALARIES is invalidated because it depends on JWARD.EMP. Then when JWARD.DEPT_SALARIES is used, Oracle attempts to recompile the view. As Oracle resolves the reference to EMP, it finds JWARD.EMP (PUBLIC.EMP

is no longer the referenced object). Because JWARD.EMP does not have a SAL column, Oracle finds errors when replacing the view, leaving it invalid.

In summary, dependencies on nonexistent objects checked during object resolution must be managed in case the nonexistent object is later created.

Shared SQL Dependency Management

In addition to managing dependencies among schema objects, Oracle also manages dependencies of each shared SQL area in the shared pool. If a table, view, synonym, or sequence is created, altered, or dropped, or a procedure or package specification is recompiled, all dependent shared SQL areas are invalidated. At a subsequent execution of the cursor that corresponds to an invalidated shared SQL area, Oracle reparses the SQL statement to regenerate the shared SQL area.

Local and Remote Dependency Management

Tracking dependencies and completing necessary recompilations are performed automatically by Oracle. In the simplest case, Oracle must manage dependencies among the objects in a single database (local dependency management). For example, a statement in a procedure can reference a table in the same database. In more complex systems, Oracle must manage dependencies in distributed environments across a network (remote dependency management). For example, an Oracle Forms trigger can depend on a schema object in the database. In a distributed database, a local view's defining query can reference a remote table.

Managing Local Dependencies

Oracle manages all local dependencies using the database's internal "depends-on" table, which keeps track of each schema object's dependent objects. When a referenced object is modified, Oracle uses the depends-on table to identify dependent objects, which are then invalidated. For example, assume a stored procedure UPDATE_SAL references the table JWARD.EMP. If the definition of the table is altered in any way, the status of every object that references JWARD.EMP is changed to INVALID, including the stored procedure UPDATE_SAL. As a result, the procedure cannot be executed until it has been recompiled and is valid. Similarly, when a DML privilege is revoked from a user, every dependent object in the user's schema is invalidated. However, an object that is invalid because authorization was revoked can be revalidated by "reauthorization", in which case it does not require full recompilation.

Managing Remote Dependencies

Application-to-database and distributed database dependencies must also be managed. For example, an Oracle Forms application might contain a trigger that references a table, or a local stored procedure might call a remote procedure in a distributed database system. The database system must account for dependencies among such objects. Oracle uses different mechanisms to manage remote dependencies, depending on the objects involved.

Dependencies Among Local and Remote Database Procedures

Dependencies among stored procedures (including functions, packages, and triggers) in a distributed database system are managed using *timestamp checking* or *signature checking.*

The dynamic initialization parameter REMOTE_DEPENDENCIES_MODE determines whether timestamps or signatures govern remote dependencies.

Additional Information:

See *Oracle8i Application Developer's Guide - Fundamentals* for details about managing remote dependencies with timestamps or signatures.

Timestamp Checking

In the timestamp checking dependency model, whenever a procedure is compiled or recompiled its *timestamp* (the time it is created, altered, or replaced) is recorded in the data dictionary. Additionally, the compiled version of the procedure contains information about each remote procedure that it references, including the remote procedure's schema, package name, procedure name, and timestamp.

When a dependent procedure is used, Oracle compares the remote timestamps recorded at compile time with the current timestamps of the remotely referenced procedures. Depending on the result of this comparison, two situations can occur:

- The local and remote procedures execute without compilation if the timestamps match.
- The local procedure is invalidated if any timestamps of remotely referenced procedures do not match, and an error is returned to the calling environment. Furthermore, all other local procedures that depend on the remote procedure with the new timestamp are also invalidated. For example, assume several local procedures call a remote procedure, and the remote procedure is recompiled. When one of the local procedures is executed and notices the different timestamp of the remote procedure, every local procedure that depends on the remote procedure is invalidated.

Actual timestamp comparison occurs when a statement in the body of a local procedure executes a remote procedure; only at this moment are the timestamps compared via the distributed database's communications link. Therefore, all statements in a local procedure that precede an invalid procedure call might execute successfully. Statements subsequent to an invalid procedure call do not execute at all (compilation is required). However, any DML statements executed before the invalid procedure call are rolled back.

Signature Checking

Oracle provides the additional capability of remote dependencies using *signatures*. The signature capability affects only remote dependencies. Local (same server) dependencies are not affected, as recompilation is always possible in this environment.

The signature of a procedure contains information about the

- name of the package, procedure, or function
- base types of the parameters
- modes of the parameters (IN, OUT, and IN OUT)

Note:

Only the types and modes of parameters are significant. The name of the parameter does not affect the signature.

If the signature dependency model is in effect, a dependency on a remote program unit (package, stored procedure, stored function, or trigger) causes an invalidation of the dependent unit if the dependent unit contains a call to a procedure in the parent unit, and the signature of this procedure has been changed in an incompatible manner.

Dependencies Among Other Remote Schema Objects

Oracle does not manage dependencies among remote schema objects other than local-procedure-to-remote-procedure dependencies.

For example, assume that a local view is created and defined by a query that references a remote table. Also assume that a local procedure includes a SQL statement that references the same remote table. Later, the definition of the table is altered.

As a result, the local view and procedure are never invalidated, even if the view or procedure is used after the table is altered, and even if the view or procedure now returns errors when used (in this case, the view or procedure must be altered manually so errors are not returned). In such cases, lack of dependency management is preferable to unnecessary recompilations of dependent objects.

Dependencies of Applications

Code in database applications can reference objects in the connected database. For example, OCI, Precompiler, and SQL*Module applications can submit anonymous PL/SQL blocks; triggers in Oracle Forms applications can reference a schema object.

Such applications are dependent on the schema objects they reference. Dependency management techniques vary, depending on the development environment. Refer to the appropriate manuals for your application development tools and your operating system for more information about managing the remote dependencies within database applications.
